FileMaker® Server 9

Custom Web Publishing mit XML und XSLT

© 2007 FileMaker, Inc. Alle Rechte vorbehalten.

FileMaker, Inc.
5201 Patrick Henry Drive
Santa Clara, California 95054, USA

FileMaker ist eine Marke von FileMaker, Inc., eingetragen in den USA und anderen Ländern. ScriptMaker und das Dateiordner-Logo sind Marken von FileMaker, Inc. Alle anderen Marken sind Eigentum der jeweiligen Besitzer.

Die FileMaker-Dokumentation ist urheberrechtlich geschützt. Sie dürfen diese Dokumentation ohne schriftliche Genehmigung von FileMaker weder vervielfältigen noch verteilen. Diese Dokumentation darf ausschließlich mit einer gültigen, lizenzierten Kopie der FileMaker-Software verwendet werden.

Alle in den Beispielen erwähnten Personen und Firmen sind rein fiktiv und jegliche Ähnlichkeit mit bestehenden Personen und Firmen ist rein zufällig. Die Danksagungen und Urheberrechtshinweise finden Sie im entsprechenden Dokument, das mit der Software geliefert wurde. Die Erwähnung von Produkten Dritter dient nur zur Information und stellt keine Empfehlung dar. FileMaker, Inc. übernimmt keine Verantwortung für die Leistung dieser Produkte.

Weitere Informationen finden Sie auf unserer Website unter www.filemaker.com.

Edition: 01

Inhalt

Über dieses Handbuch 9

Kapitel 1
Einführung in Custom Web Publishing 11
Über die Web Publishing Engine 12
Verarbeitung einer Web Publishing Engine-Anforderung 12
Custom Web Publishing mit PHP 13
Custom Web Publishing mit XML und XSLT 13
Vergleich von PHP mit XML und XSLT 14
Gründe für PHP 14
Gründe für XML und XSLT 14

Kapitel 2
Über Custom Web Publishing mit XML und XSLT 15
Erstellen dynamischer Websites mit der Web Publishing Engine 15
Erläuterungen zu Custom Web Publishing mit XML 15
Erläuterungen zu Custom Web Publishing mit XSLT 16
Über das Entwickeln von FileMaker XSLT-Stylesheets 16
Wichtige Funktionen in Custom Web Publishing mit XML und XSLT 16
Voraussetzungen für Web Publishing 17
Voraussetzungen für das Veröffentlichen einer Datenbank mit Custom Web Publishing 17
Voraussetzungen für Web-Benutzer zum Zugriff auf eine Custom Web Publishing-Lösung 18
Verbindung zum Internet oder zu einem Intranet 18
Die nächsten Schritte 18

Kapitel 3
Vorbereiten von Datenbanken für Custom Web Publishing 19
Aktivieren von Instant Web Publishing in einer Datenbank 19
Zugriff auf eine geschützte Datenbank 19
Schützen veröffentlichter Datenbanken 20
Web Server-Unterstützung für Internet-Medientypen (MIME) 21
Veröffentlichen des Inhalts von Medienfeldern im Web 21
Veröffentlichen von Medienfeldobjekten, die in einer Datenbank gespeichert sind 21
Veröffentlichen von Medienfeldobjekten, die als Dateiverweis gespeichert sind 22
Anzeige von Medienfelddaten durch Web-Benutzer 22
FileMaker-Scripts und Custom Web Publishing 22
Tipps und Überlegungen zu Scripts 23
Scriptverhalten in Custom Web Publishing-Lösungen 24

Kapitel 4
Zugriff auf XML-Daten mit der Web Publishing Engine 25
Verwenden von Custom Web Publishing mit XML 25
Unterschiede zwischen der Web Publishing Engine und FileMaker Pro XML-Import/Export 25
Wie die Web Publishing Engine aus einer Abfrage XML-Daten generiert 26
Allgemeines Verfahren beim Zugriff auf XML-Daten von der Web Publishing Engine 27
Erläuterung der URL-Syntax für XML-Daten und Medienobjekte 27
Erläuterung der URL-Syntax für XML-Daten 28
Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XML-Lösungen 28
Erläuterung von URL-Textkodierung 29
Zugriff auf XML-Daten über die Web Publishing Engine 30
Erläuterung von Namensräumen für FileMaker XML 31
Erläuterung der FileMaker-Datenbankfehlercodes 31
Abrufen der Document Type Definitions für die FileMaker-Grammatiken 31
Verwenden der fmresultset-Grammatik 32
Beschreibung der Elemente der fmresultset-Grammatik 32
Beispiel für XML-Daten in der fmresultset-Grammatik 34
Verwenden anderer FileMaker XML-Grammatiken 35
Beschreibung der Elemente in der FMPDSORESULT-Grammatik 35
Beispiel für XML-Daten in der FMPDSORESULT-Grammatik 36
Beschreibung der Elemente der Grammatik FMPXMLLAYOUT 36
XML-Daten in der FMPDSORESULT-Grammatik – Beispiel 37
Erläuterung von UTF-8-kodierten Daten 38
Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings 38
Wechseln des Layouts für eine XML-Antwort 40
Erläuterung der Verarbeitung einer XML-Abfrage 41
Verwenden von Server-seitiger und Client-seitiger Stylesheet-Verarbeitung 41
Fehlerbehebung beim XML-Dokumentzugriff 42

Kapitel 5
Einführung in Custom Web Publishing mit XSLT 43
Erläuterung von FileMaker XSLT-Stylesheets 43
Beispiele für die Verwendung von FileMaker XSLT-Stylesheets 43
Einstieg in die Verwendung von Custom Web Publishing mit XSLT 44
Wie die Web Publishing Engine auf der Basis von XML-Daten und XSLT-Stylesheets Seiten generiert 44
Allgemeine Schritte bei der Verwendung von Custom Web Publishing mit XSLT 45
Generieren von FileMaker-XSLT-Stylesheets mithilfe des FileMaker XSLT-Site-Assistenten 46
Vor der Verwendung des XSLT-Site-Assistenten 46
Starten des XSLT-Site-Assistenten 47
Verwenden des XSLT-Site-Assistenten 47
Über die vom XSLT-Site-Assistenten generierten Stylesheets 48
Verwenden von FileMaker XSLT-Stylesheets in einer Website oder einem Programm 48
Fehlerbehebung in XSLT-Stylesheets 49

Kapitel 6

Entwickeln von FileMaker XSLT-Stylesheets 51
Verwenden von XSLT-Stylesheets mit der Web Publishing Engine 51
Referenz der FileMaker XSLT-Erweiterungsfunktionen 52
Über die FileMaker XSLT-Starter-Lösung 52
Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets 52
Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen 53
Verwenden von Query-Strings in FileMaker XSLT-Stylesheets 54
Angeben einer XML-Grammatik für ein FileMaker XSLT-Stylesheet 55
Erläuterung von Namespaces und Präfixen für FileMaker XSLT-Stylesheets 55
Verwenden von statisch definierten Query-Befehlen und Query-Parametern 56
Festlegen von Textkodierung für Abfragen 57
Angeben einer Ausgabemethode und -kodierung 58
Erläuterung der Kodierung von XSLT-Stylesheets 58
Verarbeitung von XSLT-Abfragen, die nicht FileMaker Server abfragen 58
Verwendung von Tokens zur Übergabe von Informationen zwischen Stylesheets 59
Verwenden der FileMaker XSLT-Erweiterungsfunktionen und -parameter 59
- Erläuterung der FileMaker-spezifischen XSLT-Parameter, die von der Web Publishing Engine gesetzt werden 60
- Verwenden der Query-Information in einer Abfrage 60
- Abrufen von Clientinformationen 61
- Verwenden des Basis-URI-Parameters der Web Publishing Engine 61
- Verwenden des Parameters „authenticated base URI“ 62
- Laden zusätzlicher Dokumente 62
- Verwenden der Layoutinformationen für eine Datenbank in einem Stylesheet 63
- Verwenden von Inhaltspufferung 63
- Verwenden von Web Publishing Engine-Sitzungen zum Speichern von Informationen zwischen Abfragen 64
- Verwenden der Sitzungserweiterungsfunktionen 65
- Senden von E-Mail-Nachrichten von der Web Publishing Engine 66
- Verwenden von Header-Funktionen 68
- Verwenden der Cookie-Erweiterungsfunktionen 69
- Verwenden der Erweiterungsfunktionen zur Stringmanipulation 70
- Vergleich von Strings mithilfe regulärer Perl 5-Ausdrücke 71
- Prüfen von Werten in einem Feld, das als Kontrollkästchen formatiert ist 71
- Verwenden der Datums-, Zeit- und Tages-Erweiterungsfunktionen 72
- Prüfen des Fehlerstatus von Erweiterungsfunktionen 76
- Verwenden der Protokollierung 76

Verwenden der Server-seitigen Verarbeitung von Scriptingsprachen 77
- Definieren einer Erweiterungsfunktion 77
- Ein Beispiel für eine Erweiterungsfunktion 78

Kapitel 7
Einstellen, Testen und Überwachen einer Site 81
Einstellen einer Custom Web Publishing-Site 81
Testen einer Custom Web Publishing-Site 82
Beispiele für Stylesheets zum Testen der XML-Ausgabe 83
Überwachen Ihrer Website 83
Verwenden der Web-Server-Zugriffs- und Fehlerprotokolle 84
Verwenden des Web Publishing Engine-Programmprotokolls 84
Verwenden des Web-Server-Modul-Fehlerprotokolls 85
Verwenden von internen Web Publishing Core-Zugriffsprotokollen 85

Anhang A
Gültige Namen in Query-Strings 87
Erläuterung von Query-Befehlen und -Parametern 87
Richtlinien für die Verwendung von Query-Befehlen und -Parametern 88
Erläuterung der Referenz von FileMaker-Query-Strings 89
Erläuterung der Syntax für einen voll qualifizierten Feldnamen 89
Verwenden von Query-Befehlen für Ausschnittfelder 89
Erläuterung der Syntax für das Angeben eines Variablenfelds 91
Query-Befehlsreferenz 91
Query-Befehl –dbnames (Datenbanknamen) 92
Query-Befehl –delete (Datensatz löschen) 92
Query-Befehl –dup (Datensatz duplizieren) 92
Query-Befehl –edit (Datensatz bearbeiten) 92
Query-Befehle –find, –findall oder –findany (Datensätze suchen) 92
–findquery (Compound-Suche, Query-Befehl) 93
–layoutnames (Layoutnamen) (Query-Befehl) 93
–new (Neuer Datensatz) (Query-Befehl) 94
–process (XSLT-Stylesheets verarbeiten) 94
–scriptnames (Scriptnamen) (Query-Befehl) 94
Query-Befehl –view (Layoutinformationen anzeigen) 94
Query-Parameterreferenz 95
Query-Parameter –db (Datenbankname) 95
–delete.related (Löschen von Ausschnittdatensätzen, Query-Parameter) 95
Query-Parameter –encoding (XSLT-Abfrage-Kodierung) 95
Query-Parameter –field (Medienfeldname) 95
Query-Parameter „fieldname“ (Nicht-Medienfeldname) 96
Query-Parameter „fieldname.Op“ (Vergleichsoperator) 96
Query-Parameter –grammar (Grammatik für XSLT-Stylesheets) 97
Query-Parameter –lay (Layout) 97
Query-Parameter –lay.response (Layout wechseln für Antwort) 98
Query-Parameter –lop (Logischer Operator) 98
Query-Parameter –max (Maximum Datensätze) 98
Query-Parameter –modid (Änderungs-ID) 98

–query (Compound-Suchabfrage, Query-Parameter) 99
Query-Parameter –recid (Datensatz-ID) 100
–relatedsets.filter (Ausschnittdatensätze filtern, Query-Parameter) 100
–relatedsets.max (Ausschnittdatensätze beschränken, Query-Parameter) 100
–script (Script) (Query-Parameter) 101
Query-Parameter –script.param (Parameter an Script übergeben) 101
Query-Parameter –script.prefind (Script vor Suchen) 101
Query-Parameter –script.prefind.param (Parameter an Script vor Suchen übergeben) 102
Query-Parameter –script.presort (Script vor Sortieren) 102
Query-Parameter –script.presort.param (Parameter an Script vor Sortieren übergeben) 102
Query-Parameter –skip (Datensätze überspringen) 103
Query-Parameter –sortfield (Sortierfeld) 103
Query-Parameter –sortorder (Sortierfolge) 103
Query-Parameter –stylehref (Stil-Href) 104
Query-Parameter –styletype (Stilart) 105
Query-Parameter –token.[string] (Werte zwischen XSLT-Stylesheets übergeben) 105

Anhang B
Fehlercodes für Custom Web Publishing 107
Fehlercodenummern für FileMaker-Datenbanken 107
Fehlercodenummern für die Web Publishing Engine 114
Fehlercodenummern für die FileMaker XSLT-Erweiterungsfunktionen 115

Index 117

Vorwort

Über dieses Handbuch

Dieses Handbuch setzt voraus, dass Sie mit XML und XSLT, der Entwicklung von Websites und dem Einsatz von FileMaker® Pro zur Erstellung von Datenbanken vertraut sind. Sie sollten mit den Grundlagen der Datenbankgestaltung von FileMaker Pro vertraut sein und die Konzepte von Feldern, Beziehungen, Layouts, Ausschnitten und Medienfeldern verstehen. Dieses Handbuch enthält die folgenden Informationen über Custom Web Publishing mit XML und XSLT auf FileMaker Server:

- Voraussetzungen für die Entwicklung einer Custom Web Publishing-Lösung mit XML oder XSLT
- Veröffentlichen Ihrer Datenbanken mit XML oder XSLT
- Voraussetzungen für Web-Benutzer zum Zugriff auf eine Custom Web Publishing-Lösung
- Abruf von XML-Daten aus Datenbanken, die FileMaker Server bereitstellt
- Entwicklung von FileMaker XSLT-Stylesheets

Wichtig Sie können PDF-Dateien der FileMaker-Dokumentation von www.filemaker.de/downloads herunterladen. Aktualisierungen dieses Dokuments erhalten Sie ebenfalls auf der Website.

Die Dokumentation für FileMaker Server umfasst die folgenden Informationen:

Weitere Informationen über	Siehe
Installieren und Konfigurieren von FileMaker Server	*FileMaker Server Einführung* FileMaker Server Hilfe
Instant Web Publishing	*FileMaker Instant Publishing Handbuch*
Custom Web Publishing mit PHP	*FileMaker Server Custom Web Publishing mit PHP*
Verwenden des PHP-Site-Assistenten	PHP-Site-Assistent Hilfe
Custom Web Publishing mit XML und XSLT	*FileMaker Server Custom Web Publishing mit XML und XSLT* (dieses Handbuch)
Verwenden des XSLT-Site-Assistenten	XSLT-Site-Assistent Hilfe
Installation und Konfiguration der ODBC- und JDBC-Treiber und Verwendung von ODBC und JDBC	*FileMaker ODBC- und JDBC-Handbuch*
Wie FileMaker Server Auto Update das aktuellste Plugin für FileMaker Pro-Datenbank-Clientcomputer herunterladen kann	*FileMaker-Handbuch zur Aktualisierung von Plugins*

Kapitel 1
Einführung in Custom Web Publishing

Mit FileMaker Server können Sie Ihre FileMaker-Datenbank im Internet oder in einem Intranet wie folgt veröffentlichen:

Instant Web Publishing: Mit Instant Web Publishing können Sie Ihre Datenbank schnell und einfach im Web veröffentlichen. Sie müssen Ihre Datenbankdateien dazu nicht verändern oder zusätzliche Software installieren – jeder Benutzer mit einem kompatiblen Webbrowser und Zugriff auf das Internet bzw. ein Intranet kann eine Verbindung zu Ihrer Datenbank herstellen, um Datensätze anzuzeigen, zu bearbeiten, zu sortieren oder zu durchsuchen, wenn Sie ihm die entsprechenden Zugriffsrechte geben.

Für Instant Web Publishing muss der Host-Computer FileMaker Pro oder FileMaker Server ausführen. Die Benutzeroberfläche ähnelt der FileMaker Pro-Desktop-Anwendung. Die Web-Seiten und Formulare, mit denen der Web-Benutzer arbeitet, hängen von den in der FileMaker Pro-Datenbank definierten Layouts und Ansichten ab. Weitere Informationen finden Sie im FileMaker Instant Web Publishing Handbuch.

Statisches Veröffentlichen: Wenn sich Ihre Daten selten ändern oder Sie keine Live-Verbindung Ihrer Benutzer zu Ihrer Datenbank wünschen, können Sie die Daten statisch veröffentlichen. Bei der statischen Veröffentlichung exportieren Sie Daten aus einer FileMaker Pro-Datenbank, um eine Web-Seite zu erstellen, die Sie anhand von HTML weiter anpassen können. Die Web-Seite ändert sich nicht, wenn sich Informationen in Ihrer Datenbank ändern, und die Benutzer stellen keine Verbindung zu Ihrer Datenbank her. (Mit Instant Web Publishing werden die Daten in einem Webbrowser-Fenster jedes Mal aktiviert, wenn der Browser eine Anfrage an FileMaker Server sendet.) Weitere Informationen finden Sie im *FileMaker Instant Web Publishing Handbuch*.

Custom Web Publishing: Erscheinungsbild und Funktionalität Ihrer veröffentlichten Datenbank können Sie über die für FileMaker Server verfügbaren Technologien für Custom Web Publishing beeinflussen. Für FileMaker Server, der die veröffentlichten Datenbanken bereitstellt, muss weder FileMaker Pro installiert noch Custom Web Publishing verfügbar sein.

Mit Custom Web Publishing können Sie:

- Ihre Datenbank in andere Web-Sites integrieren.
- festlegen, wie Benutzer mit den Daten arbeiten.
- steuern, wie Daten in Webbrowsern angezeigt werden.

FileMaker Server stellt zwei Custom Web Publishing-Technologien zur Verfügung:

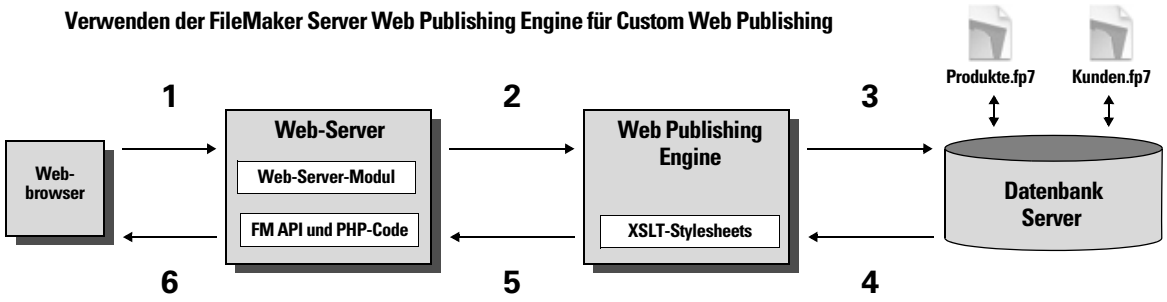
- Custom Web Publishing mit PHP: Verwenden Sie das FileMaker API für PHP, das eine objektorientierte PHP-Schnittstelle zu FileMaker Pro-Datenbanken bereitstellt, um Ihre FileMaker-Daten in einer PHP-Web-Anwendung zu integrieren. Sie können den PHP-Site-Assistenten verwenden, um eine komplette PHP-Website zu generieren, oder Ihre PHP-Web-Seiten selbst programmieren.
- Custom Web Publishing mit XML und XSLT:
  - Verwenden Sie XML Data Publishing, um FileMaker-Daten mit anderen Websites und Anwendungen auszutauschen.
  - Verwenden Sie Server-verarbeitete XSLT-Stylesheets, um beliebige Untergruppen von FileMaker-Daten in andere Websites, mit anderer Middleware und mit Spezialprogrammen zu integrieren. Sie können den XSLT-Site-Assistenten verwenden, um XSLT-Stylesheets zu generieren oder die Stylesheets selbst zu erstellen.

Über die Web Publishing Engine

Um Instant Web Publishing und Custom Web Publishing zu unterstützen, verwendet FileMaker Server einen Satz von Softwarekomponenten, die sogenannte *FileMaker Server Web Publishing Engine*. Die Web Publishing Engine verwaltet Interaktionen zwischen dem Browser eines Web-Benutzers, Ihrem Web-Server und FileMaker Server.

Custom Web Publishing mit XML und XSLT: Die Web Publishing Engine fungiert als XSLT-Prozessor und bietet Ausgaben als HTML, XML oder Text (z. B. vCards) auf dem Web-Server, der dann die Ausgabe für den Webbrowser liefert. Web-Benutzer greifen auf Ihre Custom Web Publishing-Lösung zu, indem sie auf einen HREF-Link klicken oder einen Uniform Resource Locator (URL) eingeben, der die Web-Server-Adresse und eine FileMaker-Query-Zeichenfolgenabfrage angibt. Der URL kann auf XML-Daten zugreifen oder auf ein XSLT-Stylesheet verweisen. Die Web Publishing Engine gibt die XML-Daten, die in der Query-Zeichenfolgenabfrage angegeben sind, oder das Ergebnis des referenzierten XSLT-Stylesheets zurück.

Custom Web Publishing mit PHP: Wenn ein Web-Benutzer auf Ihre Custom Web Publishing-Lösung zugreift, stellt PHP auf FileMaker Server eine Verbindung mit der Web Publishing Engine her und reagiert über das FileMaker API für PHP.



Verarbeitung einer Web Publishing Engine-Anforderung

1. Eine Anforderung wird von einem Webbrowser oder Programm an den Web-Server gesendet.
2. Der Web-Server leitet die Anforderung über das FileMaker Web-Server-Modul an die Web Publishing Engine weiter.
3. Die Web Publishing Engine fordert Daten von der Datenbank an, die der Datenbank-Server bereitstellt.
4. FileMaker Server sendet die angeforderten FileMaker-Daten an die Web Publishing Engine.
5. Die Web Publishing Engine konvertiert die FileMaker-Daten, um die Anforderung zu beantworten.
   - Für PHP-Anforderungen antwortet die Web Publishing Engine auf die API-Anforderung.
   - Für XML-Anforderungen sendet die Web Publishing Engine die XML-Daten direkt an den Web-Server.
   - Für XSLT-Abfragen verwendet die Web Publishing Engine ein XSLT-Stylesheet, um die XML-Daten zu formatieren oder zu transformieren, und generiert die Ausgabe an den Web-Server als HTML-Seiten, als XML-Dokument oder Text.
6. Der Web-Server sendet die Ausgabe an den anfordernden Webbrowser oder das anfordernde Programm.

Wichtig Sicherheitsüberlegungen sind wichtig, wenn Sie Daten im Web veröffentlichen. Sehen Sie sich die Sicherheitsrichtlinien im *FileMaker Pro Benutzerhandbuch* an, das als PDF-Datei unter www.filemaker.de/downloads verfügbar ist.

Custom Web Publishing mit PHP

Das FileMaker API für PHP bietet eine objektorientierte PHP-Schnittstelle zu FileMaker-Datenbanken. Mithilfe des FileMaker API für PHP kann sowohl auf Daten als auch auf Logik, die in einer FileMaker Pro-Datenbank gespeichert sind, zugegriffen werden. Diese können auch im Internet veröffentlicht oder an andere Anwendungen exportiert werden. Das API unterstützt zudem komplexe und zusammengesetzte Suchabfragen für das Extrahieren und Filtern von in FileMaker Pro-Datenbanken gespeicherten Daten.

Ursprünglich als prozedurale Programmiersprache entwickelt, wurde PHP als objektorientierte Programmiersprache für die Web-Entwicklung ausgebaut. PHP bietet Programmiersprachenfunktionalität für den Aufbau praktisch jeder Art von Logik innerhalb einer Site-Seite. Zum Beispiel können Sie bedingte Logikkonstrukte verwenden, um die Seitengenerierung, das Datenrouting oder den Workflow zu steuern. PHP bietet zudem Funktionen für Site-Administration und Sicherheit.

Zusätzlich können Sie den FileMaker PHP-Site-Assistenten verwenden, um PHP-Code zu erstellen, der alle erforderlichen Voraussetzungen und Funktionen beinhaltet, um korrekt auf die Daten in einer FileMaker Pro-Datenbank zuzugreifen. Der PHP-Site-Assistent generiert eine mehrseitige Website, die Web-Benutzer in die Lage versetzt, in einer Datenbank zu suchen, eine Liste von Datensätzen anzuzeigen, Datensätze zu durchblättern, zu sortieren, hinzuzufügen, zu bearbeiten, zu duplizieren, zu löschen und einen Statistikbericht anzuzeigen. FileMaker-Entwickler, die nur über geringe PHP-Erfahrung verfügen, können den PHP-Site-Assistenten verwenden, um eine ganze PHP-Website zu generieren. PHP-Entwickler, die nur über geringe Erfahrung mit FileMaker verfügen, können den PHP-Site-Assistenten verwenden, um die Objekte und Methoden des FileMaker API für PHP zu verstehen.

Custom Web Publishing mit XML und XSLT

FileMaker Custom Web Publishing mit XML ermöglicht Ihnen, Abfrageanforderungen an eine FileMaker Pro-Datenbank zu senden, die von FileMaker Server bereitgestellt wird, und die resultierenden Daten anzuzeigen und zu ändern. Mithilfe einer HTTP-Abfrage mit den geeigneten Query-Befehlen und -Parametern können Sie FileMaker-Daten als ein XML-Dokument abrufen. Sie können die XML-Daten dann in andere Programme exportieren oder ihnen ein XSLT-Stylesheet zuweisen.

FileMaker Custom Web Publishing mit XSLT ermöglicht das Umwandeln, Filtern oder Formatieren von XML-Daten für die Verwendung in Webbrowsern oder anderen Programmen. Sie können:

- ein XSLT-Stylesheet verwenden, um die Daten zwischen einer FileMaker XML-Grammatik und einer anderen XML-Grammatik für andere Programme oder andere Datenbanken umzuwandeln.
- die Daten filtern, indem Sie steuern, welche Datenbankfelder vom Stylesheet veröffentlicht werden.
- das Erscheinungsbild der Daten in einer Web-Seite formatieren und steuern, wie der Web-Benutzer mit den Daten interagiert.

Die Web Publishing Engine verwendet Ihre Stylesheets, um Daten aus einer FileMaker-Datenbank zu beziehen, wenn ein Web-Benutzer eine HTTP-Abfrage und einen URL sendet, der auf eines Ihrer XSLT-Stylesheets verweist. Die Web Publishing Engine verwendet ein Stylesheet, um die XML-Daten umzuwandeln und zu formatieren, und generiert die resultierende HTML-Seite, mit der der Web-Benutzer arbeiten kann.

Der FileMaker XSLT-Site-Assistent ist eine Anwendung, mit der Sie grundlegende XSLT-Stylesheets als Ausgangspunkt für Custom Web Publishing mit XSLT erstellen können. Der XSLT-Site-Assistent generiert Stylesheets für Seiten, die die Datenbank durchsuchen, einen Datensatz anzeigen, die Datensätze in der Datenbank auflisten, Datensätze sortieren, hinzuzufügen, bearbeiten, duplizieren, löschen und einen Statistikbericht anzeigen.

Vergleich von PHP mit XML und XSLT

Der folgende Abschnitt zeigt einige Richtlinien auf, um die beste Lösung für Ihre Site zu bestimmen.

Gründe für PHP

- PHP ist eine leistungsstarke, objektorientierte prozedurale Scriptingsprache und relativ leicht zu erlernen. Es gibt viele Ressourcen für Schulungen, Entwicklung und Support.
- Mithilfe des FileMaker API für PHP kann sowohl auf Daten als auch auf Logik, die in einer FileMaker Pro-Datenbank gespeichert sind, zugegriffen werden. Diese können auch im Internet veröffentlicht oder an andere Anwendungen exportiert werden.
- Mit PHP können Sie bedingte Logik verwenden, um den Seitenaufbau oder den Fluss zu kontrollieren.
- PHP bietet Programmiersprachenfunktionalität für den Aufbau vieler Arten von Logik innerhalb einer Site-Seite.
- PHP ist eine der beliebtesten Web-Scriptingsprachen.
- PHP ist eine Open-Source-Sprache, verfügbar unter http://php.net.
- PHP ermöglicht den Zugriff auf eine große Vielzahl von Third-Party-Komponenten, die Sie in Ihre Lösungen integrieren können.

Hinweis Weitere Informationen über Custom Web Publishing mit PHP finden Sie im Handbuch *FileMaker Server Custom Web Publishing mit PHP*.

Gründe für XML und XSLT

- Die Syntax der FileMaker XML-Abfrageparameter ist für die Datenbankinteraktion konzipiert und vereinfacht die Lösungsentwicklung.
- XML und XSLT sind W3C-Standards.
- XML ist ein von Maschinen und Menschen lesbares Format, das Unicode unterstützt, so dass Daten in jeder geschriebenen Sprache kommuniziert werden können.
- XML ist bestens geeignet für die Darstellung von Datensätzen, Listen und Daten mit Baumstruktur.
- Mit XSLT können Sie die XML-Ausgabe in strukturierte Textdokumente wie RSS, RTF oder vCard umwandeln.
- Sie können XSLT verwenden, um die XML-Ausgabe aus einer Grammatik in eine andere umzuwandeln.
- Vorlagen vereinfachen die Anwendung bedingter Formatierung auf variable Daten.
- Sie können FMPXMLRESULT-basierte Stylesheets für Custom Web Publishing und für den XML-Export aus FileMaker Pro-Datenbanken verwenden.
- FileMaker Server verwaltet die Verarbeitung von FileMaker-XSLT-Stylesheets und verhindert den unbefugten Zugriff auf Daten, die bei Verwendung von Client-seitigen XSLT-Stylesheets ungesichert sein könnten.

Kapitel 2
Über Custom Web Publishing mit XML und XSLT

Erstellen dynamischer Websites mit der Web Publishing Engine

Die Web Publishing Engine bietet Custom Web Publishing für FileMaker Server mithilfe von XML-Datenveröffentlichung und vom Server verarbeiteten XSLT-Stylesheets. Custom Web Publishing bietet mehrere Vorteile:

- **Anpassung**: Sie können festlegen, wie Web-Benutzer mit FileMaker-Daten interagieren und wie die Daten in Webbrowsern angezeigt werden.
- **Datenaustausch**: Mithilfe von FileMaker XML können Sie FileMaker-Daten mit anderen Websites und Programmen austauschen.
- **Datenintegration**: Mit FileMaker XSLT-Stylesheets können Sie FileMaker-Daten in andere Websites und mit anderen Middleware- und Spezialprogrammen integrieren. Sie können die Daten so darstellen, als würden sie zu einer anderen Website gehören, anstatt ein vollständiges FileMaker-Layout im Webbrowser anzuzeigen.
- **Sicherheit**: Der FileMaker Server-Administrator kann Instant Web Publishing, XML Web Publishing oder XSLT Web Publishing einzeln für alle Datenbanken, die der Server bereitstellt, aktivieren oder deaktivieren. Als Eigentümer der FileMaker-Datenbank können Sie den Zugriff der Web-Benutzer auf Instant Web Publishing, XML Web Publishing oder XSLT Web Publishing für jede Datenbank steuern.
- **Server-seitige Stylesheets**: Die Server-seitige Verarbeitung von XSLT-Stylesheets verhindert die unbefugte Anzeige vertraulicher Datenbankinformationen, die mit Client-seitigen Stylesheets möglich sein könnte.
- **Steuern und Filtern von veröffentlichten Daten**: Mit XSLT-Stylesheets können Sie die Daten und die Art der Datenbankinformationen, die Sie veröffentlichen wollen, steuern und filtern und damit eine unbefugte Nutzung der Datenbank verhindern. Sie können auch Metadaten ausblenden, wie z. B. Datenbank- und Feldnamen.
- **Basierend auf offenen Standards**: Ihnen stehen mehr Tools, Ressourcen und Fachpersonal für Custom Web Publishing-Lösungen zur Verfügung. Wenn Sie Standard-XML oder -XSLT bereits kennen, können Sie sofort mit der Entwicklung von Lösungen beginnen, nachdem Sie einige spezielle Details über Custom Web Publishing mit XML gelernt haben, z. B. die erforderliche URL-Syntax und Query-Parameter.

Erläuterungen zu Custom Web Publishing mit XML

Mit Custom Web Publishing mit XML können Sie Daten aus FileMaker-Datenbanken abrufen und die Daten einfach in anderen Ausgabeformaten verwenden. Mithilfe einer HTTP-Abfrage mit den geeigneten Query-Befehlen und -Parametern können Sie FileMaker-Daten als ein XML-Dokument abrufen. Sie können die XML-Daten dann in anderen Programmen verwenden oder ihnen ein XSLT-Stylesheet zuweisen Weitere Informationen finden Sie unter Kapitel 4, „Zugriff auf XML-Daten mit der Web Publishing Engine“.

Erläuterungen zu Custom Web Publishing mit XSLT

Custom Web Publishing mit XSLT ermöglicht das Umwandeln, Filtern oder Formatieren von XML-Daten für die Verwendung in einem Webbrowser oder in anderen Programmen. Sie können ein XSLT-Stylesheet verwenden, um die Daten zwischen einer FileMaker XML-Grammatik und einer anderen XML-Grammatik für den Einsatz in einem anderen Programm oder einer anderen Datenbank umzuwandeln. Sie können die Daten filtern, indem Sie steuern, welche Datenbankfelder vom Stylesheet veröffentlicht werden. Sie können das Erscheinungsbild der Daten in einer Web-Seite formatieren und steuern, wie der Web-Benutzer mit den Daten interagiert. Weitere Informationen finden Sie unter Kapitel 5, „Einführung in Custom Web Publishing mit XSLT“.

Die Web Publishing Engine verwendet Ihre Stylesheets, um Daten dynamisch aus einer FileMaker-Datenbank zu beziehen, wenn ein Web-Benutzer eine HTTP-Abfrage und einen URL sendet, der auf eines Ihrer XSLT-Stylesheets verweist. Die Web Publishing Engine verwendet ein Stylesheet, um die XML-Daten umzuwandeln und zu formatieren, und generiert die resultierende HTML-Seite, mit der der Web-Benutzer arbeiten kann.

Besuchen Sie für zusätzliche Informationen über die Verwendung von FileMaker Server Custom Web Publishing mit XML und XSLT die Website www.filemaker.de/downloads.

Über das Entwickeln von FileMaker XSLT-Stylesheets

FileMaker Server beinhaltet ein Tool zur Entwicklung von XSLT-Stylesheets. Der FileMaker XSLT-Site-Assistent ist eine Anwendung, mit der Sie grundlegende XSLT-Stylesheets als Ausgangspunkt für Custom Web Publishing mit XSLT erstellen können. Der XSLT-Site-Assistent eignet sich ideal, um den Aufbau von FileMaker XSLT-Stylesheets kennenzulernen. Sie können dann Ihr eigenes Erstellungswerkzeug für XSLT-Stylesheets verwenden, um die Stylesheets je nach Bedarf zu ändern Weitere Informationen finden Sie unter „Generieren von FileMaker-XSLT-Stylesheets mithilfe des FileMaker XSLT-Site-Assistenten“ auf Seite 46.

Hinweis FileMaker Server unterstützt XSLT 1.0 wie durch das World Wide Web Consortium definiert. Von Ihnen verwendete XSLT-Authoring-Tools müssen ein den Standards entsprechendes XSLT 1.0 erzeugen.

Wichtige Funktionen in Custom Web Publishing mit XML und XSLT

FileMaker Server Custom Web Publishing mit XML und XSLT bietet mehrere wichtige neue Funktionen:

- Datenbanken werden auf FileMaker Server bereitgestellt und FileMaker Pro muss nicht ausgeführt werden.
- Sie können die Server-seitige XSLT-Stylesheet-Verarbeitung nutzen, die mehr Sicherheit als die Client-seitige Stylesheet-Verarbeitung bietet.
- Sie können die Server-seitige Verarbeitung von JavaScript in XSLT-Stylesheets nutzen. Weitere Informationen finden Sie unter „Verwenden der Server-seitigen Verarbeitung von Scriptingsprachen“ auf Seite 77.
- Sie können mit Ihrem FileMaker XSLT-Stylesheet unbefugte die Verwendung von Query-Befehlen und Query-Parametern verhindern, indem Sie die zu verwendenden Query-Befehle, Parameter und Werte bei der Anforderung von XML-Daten statisch definieren. Weitere Informationen finden Sie unter „Verwenden von statisch definierten Query-Befehlen und Query-Parametern“ auf Seite 56.

- Wie in FileMaker Pro basiert der Zugriff auf Daten, Layouts und Felder auf den Benutzerkontoeinstellungen, die in den Zugriffsrechten der Datenbank definiert sind. Die Web Publishing Engine unterstützt auch mehrere andere Sicherheitsverbesserungen. Weitere Informationen finden Sie unter „Schützen veröffentlichter Datenbanken“ auf Seite 20.
- Web-Benutzer können komplexe Scripts mit mehreren Schritten ausführen. Im Web werden ungefähr 70 ScriptMaker-Scriptschritte unterstützt. Informationen finden Sie im Abschnitt „FileMaker-Scripts und Custom Web Publishing“ auf Seite 22.
- Sie können einen Parameterwert an ein FileMaker-Script übergeben. Weitere Informationen finden Sie unter „Query-Parameter –script.param (Parameter an Script übergeben)“ auf Seite 101, „Query-Parameter –script.prefind.param (Parameter an Script vor Suchen übergeben)“ auf Seite 102 und „Query-Parameter –script.presort.param (Parameter an Script vor Sortieren übergeben)“ auf Seite 102.
- Mit der fmresultset-XML-Grammatik können Sie per Namen auf Felder zugreifen und relatedset-(Ausschnitt) Daten manipulieren.
- Mithilfe von Sitzungsfunktionen in einem XSLT-Stylesheet können Sie die Informationen und Transaktionen eines Web-Benutzers in vom Server geführten Sitzungen speichern.
- Für den Zugriff auf Daten in einer Datenbank müssen Sie ein Layout angeben. Weitere Informationen finden Sie unter Anhang A, „Gültige Namen in Query-Strings“.
- Jeder Web-Benutzer kann über einen eindeutigen Variablenfeldwert verfügen, der für die Dauer der Sitzung bestehen bleibt. Allgemeine Informationen zu Variablenfeldern finden Sie in der FileMaker Pro Hilfe. Informationen über die Verwendung von Variablenfeldern mit Custom Web Publishing finden Sie unter „Erläuterung der Syntax für das Angeben eines Variablenfelds“ auf Seite 91.

Voraussetzungen für Web Publishing

Voraussetzungen für das Veröffentlichen einer Datenbank mit Custom Web Publishing

Um Datenbanken mithilfe von Custom Web Publishing mit XML oder XSLT zu veröffentlichen, benötigen Sie:

- einen FileMaker Server-Einsatz, der Folgendes umfasst:
  - einen Web-Server, entweder Microsoft IIS (Windows) oder Apache (Mac OS X)
  - den FileMaker Datenbank-Server, aktiviert für Custom Web Publishing
  - die Web Publishing Engine, installiert und konfiguriert
- eine oder mehrere FileMaker Pro-Datenbanken, die FileMaker Server bereitstellt
- IP-Adresse oder Domänenname des Hosts, auf dem der Web-Server läuft
- einen beliebigen Webbrowser und Zugriff auf den Web-Server, um Ihre Custom Web Publishing-Lösung zu entwickeln und zu testen.

Weitere Informationen finden Sie im Handbuch *FileMaker Pro Einführung*.

Voraussetzungen für Web-Benutzer zum Zugriff auf eine Custom Web Publishing-Lösung

Für den Zugriff auf eine Custom Web Publishing-Lösung mit XML oder XSLT benötigen Web-Benutzer:

- einen Webbrowser
- Zugang zum Internet oder zu einem Intranet und zum Web-Server
- IP-Adresse oder Domänenname des Hosts, auf dem der Web-Server läuft.

Wenn die Datenbank passwortgeschützt ist, müssen Web-Benutzer auch einen Benutzernamen und ein Passwort für ein Datenbankkonto eingeben.

Verbindung zum Internet oder zu einem Intranet

Wenn Sie Datenbanken im Internet oder in einem Intranet veröffentlichen, muss auf dem Host-Computer FileMaker Pro oder FileMaker Server laufen und die freizugebenden Datenbanken müssen bereitgestellt und verfügbar sein. Zusätzlich:

- Veröffentlichen Sie Ihre Datenbank auf einem Computer, der über eine ständige Verbindung zum Internet bzw. Intranet verfügt. Sie können Datenbanken zwar ohne ständige Verbindung veröffentlichen, aber sie stehen Web-Benutzern nur zur Verfügung, wenn Ihr Computer mit dem Internet bzw. Intranet verbunden ist.
- Der Host-Computer für den Web-Server, der Teil des FileMaker Server-Einsatzes ist, muss über eine eigene *statische* (permanente) IP-Adresse oder einen Domänennamen verfügen. Wenn Sie die Verbindung zum Internet über einen Internet-Anbieter (ISP) herstellen, könnte Ihre IP-Adresse *dynamisch zugewiesen* werden (d. h., sie ändert sich bei jeder Verbindung). Mit einer dynamischen IP-Adresse ist es für Web-Benutzer schwieriger, Ihre Datenbanken zu finden. Wenn Sie sich über Ihre Zugangsart nicht sicher sind, wenden Sie sich an Ihren ISP- oder Netzwerkadministrator.

Die nächsten Schritte

Hier einige Vorschläge, um mit der Entwicklung von Custom Web Publishing-Lösungen zu beginnen:

- Falls noch nicht geschehen, aktivieren Sie Custom Web Publishing mithilfe der FileMaker Server Admin Console. Informationen hierzu finden Sie in der *FileMaker Server Hilfe* und im Handbuch *FileMaker Server Einführung*.
- Öffnen Sie in FileMaker Pro jede FileMaker-Datenbank, die Sie veröffentlichen wollen, und stellen Sie sicher, dass in der Datenbank die geeigneten erweiterten Zugriffsrechte für Custom Web Publishing aktiviert sind. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 19.
- Wie Sie mithilfe von XML auf Daten in FileMaker-Datenbanken zugreifen, erfahren Sie in Kapitel 4, „Zugriff auf XML-Daten mit der Web Publishing Engine“.
- Wie Sie mit der Entwicklung von FileMaker XSLT-Stylesheets beginnen, lesen Sie in Kapitel 5, „Einführung in Custom Web Publishing mit XSLT“.

Kapitel 3
Vorbereiten von Datenbanken für Custom Web Publishing

Bevor Sie Custom Web Publishing mit einer Datenbank verwenden können, müssen Sie die Datenbank vorbereiten und vor unbefugtem Zugriff schützen.

Aktivieren von Instant Web Publishing in einer Datenbank

Sie müssen Custom Web Publishing in jeder Datenbank aktivieren, die Sie veröffentlichen wollen. Sie können Custom Web Publishing mit XML oder Custom Web Publishing mit XSLT in jeder Datenbank einzeln oder beide Technologien gleichzeitig aktivieren. Wenn Sie in einer Datenbank keine dieser Technologien aktivieren, können Web-Benutzer Custom Web Publishing nicht für den Zugriff auf die Datenbank verwenden, selbst wenn sie von FileMaker Server bereitgestellt wird, der zur Unterstützung einer Web Publishing Engine konfiguriert ist.

So aktivieren Sie Custom Web Publishing in einer Datenbank:

1. Öffnen Sie in FileMaker Pro die zu veröffentlichende Datenbank mit einem Konto, das über die Berechtigung für vollen Zugriff verfügt. Alternativ können Sie die Datenbank mit einem Konto öffnen, das über die Berechtigung „Erweiterte Zugriffsrechte verwalten“ verfügt.
2. Weisen Sie einer oder mehreren Berechtigungen eine oder beide dieser erweiterten Zugriffsrechte zu:
   - Verwenden Sie für Custom Web Publishing mit XML dieses Schlüsselwort: fmxml
   - Verwenden Sie für Custom Web Publishing mit XSLT dieses Schlüsselwort: fmxslt

   Seit FileMaker Pro Version 8 sind die Schlüsselwörter fmxml und fmxslt im Register „Erweiterte Zugriffsrechte“ für Sie definiert.
3. Weisen Sie die Berechtigung(en), die die erweiterten Custom Web Publishing-Zugriffsrechte enthalten, einem oder mehreren Konten bzw. dem Admin- oder Gastkonto zu.

Hinweis Wenn Sie Kontonamen und Passwörter für Custom Web Publishing-Lösungen definieren, verwenden Sie druckbare ASCII-Zeichen. Beispiel: a-z, A-Z und 0-9. Verwenden Sie für sicherere Kontonamen und Passwörter zudem Satzzeichen wie „!“ und „%“, aber verwenden Sie keine Doppelpunkte. Informationen über das Einrichten von Konten finden Sie in der FileMaker Pro Hilfe.

Zugriff auf eine geschützte Datenbank

Wenn Web-Benutzer eine Custom Web Publishing-Lösung für den Zugriff auf eine Datenbank verwenden, werden sie eventuell aufgefordert, ihre Kontoinformationen einzugeben. Wenn das Gastkonto für die Datenbank deaktiviert ist oder über keine Berechtigung mit dem erweiterten Zugriffsrecht für Custom Web Publishing verfügt, verwendet die Web Publishing Engine die HTTP-Standardauthentifizierung, um entsprechende Informationen von Web-Benutzern zu verlangen. Der Browser des Web-Benutzers zeigt das Dialogfeld für die HTTP-Standardauthentifizierung an, in dem der Benutzer einen Benutzernamen und ein Passwort für ein Konto eingibt, das über ein erweitertes Zugriffsrecht für Custom Web Publishing verfügt.

Die folgende Liste fasst die Abläufe zusammen, die auftreten, wenn ein Web-Benutzer eine Custom Web Publishing-Lösung für den Zugriff auf eine Datenbank verwendet:

- Wenn Sie einem Konto kein Passwort zugewiesen haben, geben Web-Benutzer nur den Kontonamen an.
- Wenn das Gastkonto deaktiviert ist, werden die Benutzer beim Zugriff auf die Datenbank aufgefordert, Kontoname und Passwort anzugeben. Für das Konto muss ein erweitertes Zugriffsrecht für Custom Web Publishing aktiviert sein.
- Wenn das Gastkonto aktiviert ist und über eine Berechtigung mit einem erweiterten Zugriffsrecht für Custom Web Publishing verfügt, öffnen automatisch alle Web-Benutzer die Datenbank mit den Zugriffsrechten, die dem Gastkonto zugewiesen sind. Wenn das erweiterte Zugriffsrecht für Custom Web Publishing dem Gastkonto zugewiesen ist:
  - Web-Benutzer werden nicht aufgefordert, Kontoname und Passwort einzugeben, wenn sie eine Datei öffnen.
  - Alle Web-Benutzer werden automatisch mit dem Gastkonto angemeldet und erhalten die Gast-Zugriffsrechte. Sie können Benutzern über den Scriptschritt „Erneut anmelden“ erlauben, ihre Anmeldekonten von einem Webbrowser aus zu wechseln (z. B. vom Gastkonto zu einem anderen Konto mit mehr Rechten).
  - Die Standardberechtigungen für Gastkonten umfassen den Nur-Lese-Zugriff. Sie können für dieses Konto die Standardberechtigungen einschließlich erweiterten Zugriffsrechten ändern. Weitere Informationen finden Sie in der FileMaker Pro Hilfe.

Hinweis Standardmäßig können Web-Benutzer ihr Kontopasswort nicht von einem Webbrowser aus ändern. Sie können diese Funktion über den Scriptschritt „Passwort ändern“ in Ihre Datenbank integrieren, damit Web-Benutzer ihr Passwort vom Browser aus ändern können. Weitere Informationen finden Sie in der FileMaker Pro Hilfe.

Schützen veröffentlichter Datenbanken

Wenn Sie Custom Web Publishing mit XML oder XSLT verwenden, können Sie beschränken, wer auf Ihre veröffentlichten Datenbanken zugreifen kann.

- Weisen Sie Datenbankkonten, die für Custom Web Publishing verwendet werden, Passwörter zu.
- Aktivieren Sie Custom Web Publishing mit XML oder XSLT nur in den Berechtigungen für Konten, denen Sie den Zugriff auf Ihre veröffentlichten Datenbanken erlauben wollen.
- Um einen Typ der Custom Web Publishing-Technologie für eine einzelne Datenbank zu aktivieren oder zu deaktivieren, wählen Sie das entsprechende erweiterte Zugriffsrecht.
- Aktivieren oder deaktivieren Sie einen Typ der Custom Web Publishing-Technologie für alle Custom Web Publishing-Lösungen in der Web Publishing Engine mit der FileMaker Server Admin Console. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

- Konfigurieren Sie Ihren Web-Server, um die IP-Adressen einzuschränken, die über die Web Publishing Engine auf Ihre Datenbanken zugreifen können. Beispielsweise können Sie angeben, dass nur Web-Benutzer von der IP-Adresse 192.168.100.101 auf Ihre Datenbanken zugreifen können. Informationen über das Einschränken von IP-Adressen finden Sie in der Dokumentation zu Ihrem Web-Server.
- Verwenden Sie die Secure Sockets Layer- (SSL) Verschlüsselung für die Kommunikation zwischen Ihrem Web-Server und den Browsern der Web-Benutzer. Die SSL-Verschlüsselung wandelt Informationen, die zwischen Servern und Clients übertragen werden, mithilfe von mathematischen Formeln in unverständliche Informationen um. Der englische Fachausdruck für diese Chiffrier-Algorithmen ist Ciphers. Diese Algorithmen nutzt der Empfänger, um mithilfe von Schlüsseln, den sogenannten Chiffrierschlüsseln, die Informationen wieder in verständliche Daten umzuwandeln. Informationen über das Aktivieren und Konfigurieren von SSL erhalten Sie in der Dokumentation zu Ihrem Web-Server.

Weitere Informationen zur Sicherung Ihrer Datenbank finden Sie im *FileMaker Pro Benutzerhandbuch*, das als PDF-Datei unter www.filemaker.de/downloads verfügbar ist.

Web Server-Unterstützung für Internet-Medientypen (MIME)

Ihr Web-Server bestimmt die Unterstützung für die aktuellen MIME-Typen (MIME; Multipurpose Internet Mail Extensions), die für das Internet registriert sind. Die Web Publishing Engine ändert die MIME-Unterstützung eines Web-Servers nicht. Weitere Informationen finden Sie in der Dokumentation zu Ihrem Web-Server.

Veröffentlichen des Inhalts von Medienfeldern im Web

Der Inhalt eines Medienfelds, wie z. B. eine Bilddatei, kann in einer FileMaker-Datenbank oder als Dateiverweis mit einem relativen Pfad gespeichert werden.

Hinweis Die Web Publishing Engine unterstützt kein Streaming von Filmen. Web-Benutzer müssen eine ganze Filmdatei herunterladen, bevor sie den Film ansehen können.

Veröffentlichen von Medienfeldobjekten, die in einer Datenbank gespeichert sind

Wenn ein Medienfeld die tatsächlichen Dateien in der FileMaker-Datenbank speichert, ist keine Aktion mit dem Inhalt des Medienfelds erforderlich, sofern die Datenbankdatei korrekt auf dem FileMaker Server freigegeben und zugänglich ist. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XML-Lösungen“ auf Seite 28, und „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen“ auf Seite 53.

Veröffentlichen von Medienfeldobjekten, die als Dateiverweis gespeichert sind

Wenn ein Medienfeld Dateiverweise anstelle tatsächlicher Dateien speichert, müssen Sie folgende Schritte ausführen, um die Medienfeldobjekte zu veröffentlichen.

Hinweis Alle QuickTime-Filme werden als Verweis in einem Medienfeld gespeichert.

So veröffentlichen Sie Medienfeldobjekte, die als Dateiverweise gespeichert sind:

1. Speichern Sie die Medienobjektdateien im Web-Ordner des FileMaker Pro-Ordners.
2. Fügen Sie in FileMaker Pro die Objekte in das Medienfeld ein und wählen Sie die Option Nur Verweis auf die Datei speichern.
3. Kopieren oder verschieben Sie die Objektdateien, auf die verwiesen wird, im Web-Ordner an die gleiche Stelle im relativen Pfad des Root-Ordners der Web-Server-Software.
   - IIS: Verschieben Sie die Dateien in: <*Laufwerk*>:\Inetpub\wwwroot
   - Apache: Verschieben Sie die Dateien in: /Library/WebServer/Documents

Hinweis Für Medienobjekte, die als Dateiverweise gespeichert sind, muss Ihr Web-Server so konfiguriert sein, dass er die MIME-Typen für alle Arten von Dateien unterstützt, die Sie anbieten wollen (z. B. Filme). Weitere Informationen finden Sie in der Dokumentation zu Ihrem Web-Server.

Anzeige von Medienfelddaten durch Web-Benutzer

Wenn Sie eine Datenbank mit der Web Publishing Engine im Web veröffentlichen, können Web-Benutzer wie folgt eingeschränkt mit Daten in Medienfeldern arbeiten:

- Web-Benutzer können keine Töne abspielen oder OLE-Objekte in einem Medienfeld anzeigen – stattdessen wird eine Grafik angezeigt.
- Web-Benutzer können den Inhalt eines Medienfelds nicht ändern oder ihm Inhalte hinzufügen. Web-Benutzer können ein Medienfeld nicht verwenden, um Daten in die Datenbank hochzuladen.
- Falls Ihre Datenbank Grafiken enthält, die nicht in GIF- oder JPEG-Format sind, erstellt die Web Publishing Engine ein temporäres JPEG-Bild, wenn die Grafikdaten von einem Webbrowser angefordert werden.

FileMaker-Scripts und Custom Web Publishing

Die Funktion ScriptMaker™ in FileMaker Pro kann sich häufig wiederholende Aufgaben automatisieren oder mehrere Aufgaben zusammenzufassen. In Kombination mit Custom Web Publishing ermöglichen FileMaker-Scripts den Web-Benutzern, mehrere Aufgaben oder eine Reihe von Aufgaben durchzuführen.

FileMaker unterstützt ungefähr 70 Scriptschritte in Custom Web Publishing. Web-Benutzer können eine Vielzahl automatisierter Aufgaben ausführen, wenn Sie Scripts in einer Query-Zeichenfolge für einen URL oder in einer <?xslt–cwp–query?>-Verarbeitungsanweisung in einem XSLT-Stylesheet verwenden. Um nicht unterstützte Scriptschritte anzuzeigen, wählen Sie im Dialogfeld „Script bearbeiten“ in FileMaker Pro die Option Web-Kompatibilität anzeigen. Grau dargestellte Scripts werden nicht im Web unterstützt. Informationen über das Erstellen von Scripts finden Sie in der FileMaker Pro Hilfe.

Tipps und Überlegungen zu Scripts

Auch wenn viele Scriptschritte im Web identisch funktionieren, gibt es einige, die sich anders verhalten. Weitere Informationen finden Sie unter „Scriptverhalten in Custom Web Publishing-Lösungen“ auf Seite 24. Testen Sie alle Scripts, die von einem Webbrowser ausgeführt werden, bevor Sie Ihre Datenbank freigeben. Stellen Sie sicher, dass Sie sich mit unterschiedlichen Benutzerkonten anmelden, damit Sie sichergehen können, dass die Scripts für alle Clients gleichermaßen funktionieren. Prüfen Sie die Protokolldatei der Web Publishing Engine (pe_application_log.txt) auf Scripting-bezogene Fehler. Weitere Informationen finden Sie unter „Verwenden des Web Publishing Engine-Programmprotokolls“ auf Seite 84.

Beachten Sie folgende Tipps und Überlegungen:

- Verwenden Sie Konten und Zugriffsrechte, um die Scripts einzuschränken, die ein Web-Benutzer ausführen kann. Überprüfen Sie, ob die Scripts nur Web-kompatible Scriptschritte enthalten und nur Zugang zu Scripts gewähren, die von einem Webbrowser aus ausgeführt werden sollen.
- Berücksichtigen Sie die Nebeneffekte von Scripts, die eine Reihe von Scriptschritten ausführen, die durch Zugriffsrechte kontrolliert werden. Wenn ein Script beispielsweise einen Scriptschritt zum Löschen von Datensätzen enthält und der Web-Benutzer sich nicht mit einem Konto anmeldet, das das Löschen von Datensätzen zulässt, führt das Script den Scriptschritt zum Löschen von Datensätzen nicht aus. Das Script könnte jedoch weiter ausgeführt werden, so dass unerwartete Ergebnisse auftreten können.
- Wählen Sie im ScriptMaker-Dialogfeld „Script bearbeiten“ Script mit vollen Zugriffsrechten ausführen, damit Scripts Aufgaben ausführen können, für die Sie dem Benutzer an sich keinen Zugang gewähren wollen. Beispielsweise können Sie Benutzern untersagen, mit ihren Konten und Zugriffsrechten Datensätze zu löschen, ihnen aber die Ausführung eines Scripts gestatten, das unter vordefinierten Bedingungen innerhalb eines Scripts bestimmte Arten von Datensätzen löscht.
- Wenn Ihre Scripts Schritte enthalten, die nicht unterstützt werden (z. B. Schritte, die nicht Web-kompatibel sind), verwenden Sie den Scriptschritt AnwenderAbbruchZulassen setzen, um festzulegen, wie nachfolgende Schritte gehandhabt werden.
  - Wenn der Scriptschritt AnwenderAbbruchZulassen setzen aktiviert (ein) ist, hindern nicht unterstützte Scriptschritte das Script an der Fortsetzung.
  - Wenn AnwenderAbbruchZulassen setzen ausgeschaltet ist, werden nicht unterstützte Scriptschritte übersprungen und das Script setzt die Ausführung fort.
  - Wenn dieser Scriptschritt nicht enthalten ist, werden Scripts ausgeführt, als ob die Funktion aktiv wäre, d. h., nicht unterstützte Scriptschritte stoppen Scripts.
- Einige Scripts, die mit einem Schritt von einem FileMaker Pro-Client aus funktionieren, benötigen eventuell einen zusätzlichen Schritt Schreibe Änderung Datens./Abfrage, um die Daten auf dem Host zu speichern. Da Web-Benutzer nicht über eine direkte Verbindung zum Host verfügen, werden sie nicht sofort benachrichtigt, wenn sich Daten ändern. Funktionen wie bedingte Wertelisten funktionieren z. B. nicht gleichermaßen für Web-Benutzer, da die Daten auf dem Host gespeichert werden müssen, bevor sich die Wirkung in einem Wertelistenfeld zeigt.

- Scripts, die Daten verändern, sollten den Schritt Schreibe Änderung Datens./Abfrage enthalten, da Datenänderungen erst im Browser sichtbar werden, wenn die Daten gespeichert bzw. am Server „bestätigt“ wurden. Dies gilt für einige Scriptschritte wie Ausschneiden, Kopieren, Einfügen usw. Viele Aktionen mit nur einem Scriptschritt sollten in Scripts umgewandelt werden, die den Schritt Schreibe Änderung Datens./Abfrage enthalten. Wenn Sie Scripts entwerfen, die von einem Webbrowser aus ausgeführt werden, nehmen Sie den Schritt Schreibe Änderung Datens./Abfrage am Ende des Scripts auf, um sicherzustellen, dass alle Änderungen gespeichert werden.
- Um von der Art des Clients abhängige Scripts zu erstellen, verwenden Sie die Funktion Hole(ProgrammVersion). Wenn der zurückgegebene Wert „Web Publishing Engine“ enthält, wissen Sie, dass der aktuelle Benutzer über Custom Web Publishing auf Ihre Datenbank zugreift. Weitere Informationen zu Funktionen finden Sie in der FileMaker Pro Hilfe.
- Wenn Sie ein Script in einem XSLT-Stylesheet benutzen, das einen Status festlegt oder ändert, müssen Sie mithilfe von FileMaker Server Admin Console die Option XSLT-Datenbanksitzungen für die Web Publishing Engine aktivieren. Andernfalls werden Status zwischen Abfragen nicht beibehalten. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

Scriptverhalten in Custom Web Publishing-Lösungen

Die nachfolgenden Scriptschritte arbeiten im Web anders als in FileMaker Pro. Weitere Informationen zu allen Scriptschritten finden Sie in der FileMaker Pro Hilfe.

Scriptschritt	Verhalten in Custom Web Publishing-Lösungen
Script ausführen	Scripts können nur dann in anderen Dateien ausgeführt werden, wenn die Dateien auf FileMaker Server bereitgestellt werden und Custom Web Publishing in den anderen Dateien aktiviert ist.
Programm beenden	Meldet Web-Benutzer ab, schließt die Fenster, beendet aber den Webbrowser nicht.
AnwenderAbbruch Zulassen setzen	Legt fest, wie nicht unterstützte Scriptschritte gehandhabt werden. Aktivieren, damit Scripts gestoppt werden; deaktivieren, damit nicht unterstützte Schritte übersprungen werden. Weitere Informationen finden Sie unter „Tipps und Überlegungen zu Scripts“ auf Seite 23. Web-Benutzer können Custom Web Publishing-Scripts nicht abbrechen, aber diese Option ermöglicht, dass nicht unterstützte Scriptschritte die Fortsetzung des Scripts stoppen.
Fehleraufzeichnung setzen	Diese Option ist bei Custom Web Publishing immer aktiviert. Web-Benutzer können Custom Web Publishing-Scripts nicht abbrechen.
Scriptpause setzen	Dieses Script wird zwar in Custom Web Publishing unterstützt, sollte aber nicht benutzt werden. Wenn der Scriptschritt „Scriptpause setzen“ ausgeführt wird, hält das Script an. Nur ein Script, das erneut den Schritt „Scriptpause setzen“ enthält, kann die Ausführung fortsetzen. Wenn das Script im Pausezustand bleibt, bis die Sitzung endet, wird das Script nicht vollständig ausgeführt.
Datensätze sortieren	Eine festgelegte Sortierfolge muss mit dem Scriptschritt gespeichert werden, um mit Custom Web Publishing ausgeführt zu werden.
URL öffnen	Dieser Scriptschritt hat keine Wirkung in einer Custom Web Publishing-Lösung.
Gehe zu Feld	Im Webbrowser können Sie „Gehe zu Feld“ nicht verwenden, um zu einem bestimmten Feld zu wechseln. Sie können diesen Scriptschritt jedoch in Kombination mit anderen Scriptschritten verwenden, um Aufgaben auszuführen. Beispielsweise können Sie zu einem Feld gehen und den Inhalt kopieren und dann zu einem anderen Feld gehen und den Wert einfügen. Um die Wirkung im Browser zu sehen, müssen Sie den Datensatz mit dem Scriptschritt „Datensatz bestätigen“ speichern.
Schreibe Änderung Datens./Abfrage	Sendet den Datensatz zur Datenbank.

Kapitel 4
Zugriff auf XML-Daten mit der Web Publishing Engine

Sie können FileMaker-Daten in Extensible Markup Language- (XML) Format mithilfe der Web Publishing Engine beziehen und aktualisieren. So wie HTML zur Standardbeschreibungssprache für die Kommunikation im World Wide Web wurde, hat sich XML zur Standardsprache für den strukturierten Datenaustausch entwickelt. Diese Sprache wird von zahlreichen Einzelpersonen, Organisationen und Unternehmen zur Übermittlung von Produktinformationen, Transaktionen, Inventardaten und anderen geschäftlichen Informationen genutzt.

Verwenden von Custom Web Publishing mit XML

Wenn Sie Standard-XML bereits kennen, können Sie sofort mit dem Einsatz der Web Publishing Engine beginnen, nachdem Sie einige spezielle Details über Custom Web Publishing mit XML, wie z. B. die erforderliche URL-Syntax und Query-Parameter, gelernt haben.

Indem Sie HTTP-URL-Abfragen mit FileMaker-Query-Befehlen und -Parametern verwenden, können Sie eine Datenbank abfragen, die von FileMaker Server bereitgestellt wird, und die resultierenden Daten in XML-Format herunterladen. Sie können eine Datenbank beispielsweise auf alle Datensätze in einem bestimmten Postleitzahlenbereich abfragen und die resultierenden XML-Daten wie gewünscht verwenden.

Sie können auch die Server-seitigen XSLT-Stylesheets der Web Publishing Engine verwenden, um die XML-Daten zu filtern, die Daten in HTML oder Text wie vCards umzuformatieren oder die Daten in andere XML-Grammatiken wie Scalable Vector Graphics (SVG) umzuwandeln. Weitere Informationen finden Sie unter Kapitel 5, „Einführung in Custom Web Publishing mit XSLT“ und Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Allgemeine Informationen zu XML, weitere Beispiele, in denen es verwendet wird, sowie Links zu XML-Ressourcen finden Sie auf der FileMaker-Website unter www.filemaker.de.

Hinweis Die von der Web Publishing Engine generierten XML-Daten sind wohlgeformt und richten sich nach der XML 1.0-Spezifikation. Informationen zu den Anforderungen des XML-Aufbaus finden Sie in der XML-Spezifikation unter www.w3.org.

Unterschiede zwischen der Web Publishing Engine und FileMaker Pro XML-Import/Export

Die Web Publishing Engine und FileMaker Pro gestatten Ihnen, XML-Daten mit FileMaker-Datenbanken zu verwenden. Es gibt jedoch einige wesentliche Unterschiede zwischen den beiden Methoden:

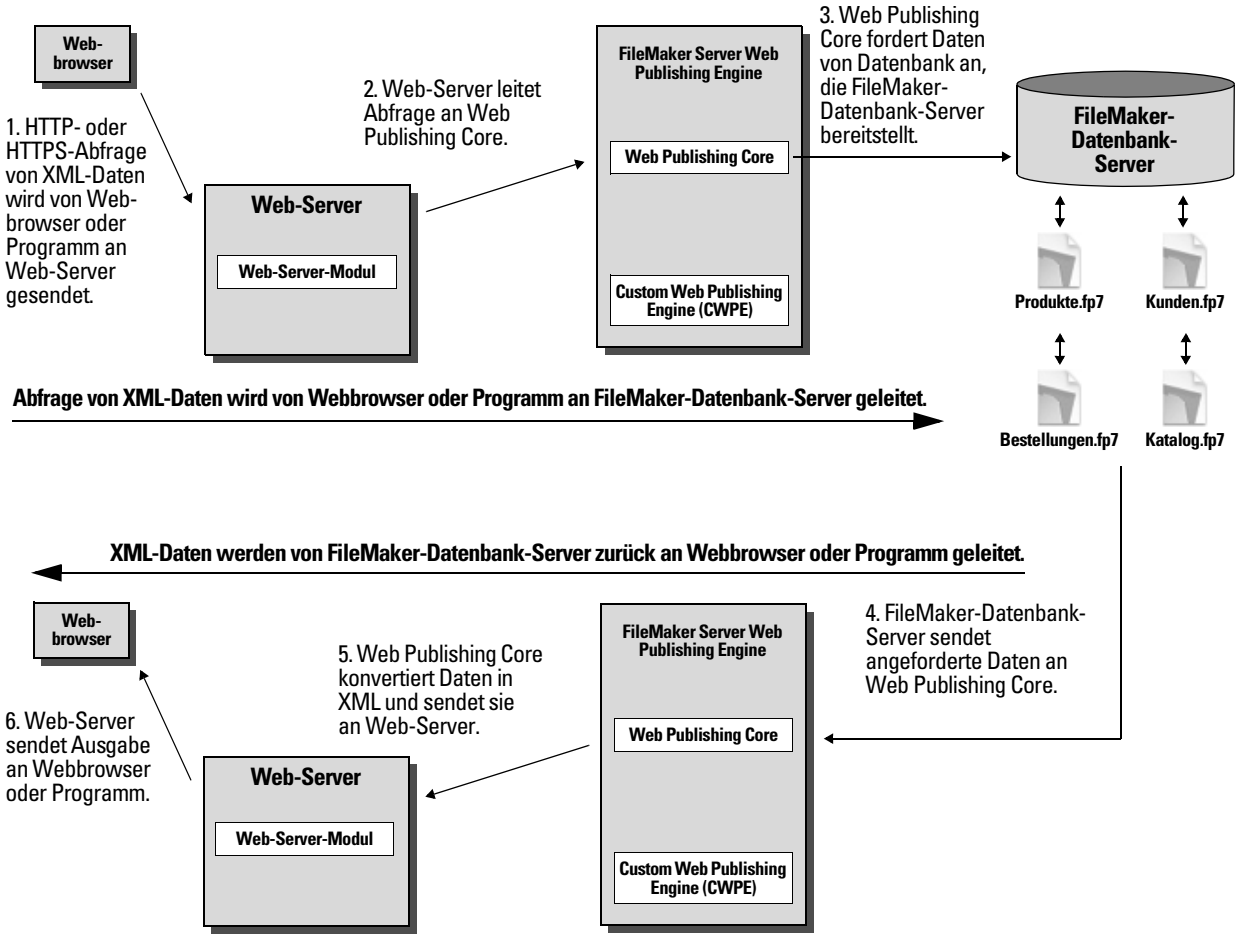
- Für den Zugriff auf XML-Daten und XSLT-Web Publishing unterstützt die Web Publishing Engine die fmresultset-, die FMPXMLRESULT- und die FMPXMLLAYOUT-Grammatik. Für XML-Import verwendet FileMaker Pro die FMPXMLRESULT-Grammatik und für den Export die FMPXMLRESULT- oder FMPDSORESULT-Grammatik. Weitere Informationen finden Sie unter „Zugriff auf XML-Daten über die Web Publishing Engine“ auf Seite 30.
- Um mit der Web Publishing Engine auf XML-Daten zuzugreifen, verwenden Sie einen Web Publishing Engine-Query-String in einem URL. Für den Import und Export von XML mit FileMaker Pro verwenden Sie FileMaker Pro-Menübefehle oder Scripts.

- Die Web Publishing Engine ist serverbasiert und kann auf demselben oder einem anderen Host als FileMaker Server installiert werden. FileMaker Pro XML-Import und -Export ist desktopbasiert.
- Sie können dynamisch auf XML-Daten von FileMaker-Datenbanken zugreifen, indem Sie URL-Abfragen mit der Web Publishing Engine verwenden. Die XML-Exportfunktion von FileMaker Pro erzeugt eine zuvor angegebene XML-Datendatei.
- Das Arbeiten mit XML-Daten über die Web Publishing Engine ist ein interaktiver Vorgang. FileMaker Pro XML-Import und -Export erfolgt im Stapelbetrieb.
- Die Web Publishing Engine kann auf XML-Daten von einem FileMaker-Ausschnitt zugreifen, FileMaker Pro nicht.
- Die Web Publishing Engine kann auf Daten in einem Medienfeld zugreifen, FileMaker Pro nicht.
- Die Web Publishing Engine bietet Echtzeitzugriff auf FileMaker-Daten über HTTP oder HTTPS, FileMaker Pro nicht.

Hinweis Informationen über das Verwenden von FileMaker Pro für Import und Export von Daten in XML-Format finden Sie in der FileMaker Pro Hilfe.

Wie die Web Publishing Engine aus einer Abfrage XML-Daten generiert

Nachdem eine Abfrage von XML-Daten an den Web-Server gesendet wurde, fragt die Web Publishing Engine die FileMaker-Datenbank ab und gibt die Daten als XML-Dokument zurück.


Web-browser
1. HTTP- oder HTTPS-Abfrage von XML-Daten wird von Web-browser oder Programm an Web-Server gesendet.
Web-Server
Web-Server-Modul
2. Web-Server leitet Abfrage an Web Publishing Core.
FileMaker Server Web Publishing Engine
Web Publishing Core
Custom Web Publishing Engine (CWPE)
3. Web Publishing Core fordert Daten von Datenbank an, die FileMaker-Datenbank-Server bereitstellt.
FileMaker-Datenbank-Server
Produkte.fp7
Kunden.fp7
Bestellungen.fp7
Katalog.fp7
Abfrage von XML-Daten wird von Webbrowser oder Programm an FileMaker-Datenbank-Server geleitet.
XML-Daten werden von FileMaker-Datenbank-Server zurück an Webbrowser oder Programm geleitet.
4. FileMaker-Datenbank-Server sendet angeforderte Daten an Web Publishing Core.
5. Web Publishing Core konvertiert Daten in XML und sendet sie an Web-Server.
6. Web-Server sendet Ausgabe an Webbrowser oder Programm.

Allgemeines Verfahren beim Zugriff auf XML-Daten von der Web Publishing Engine

Im Folgenden erhalten Sie einen Überblick über das Verfahren beim Verwenden der Web Publishing Engine für den Zugriff auf XML-Daten in einer FileMaker-Datenbank:

1. Stellen Sie in FileMaker Server Admin Console sicher, dass XML Publishing aktiviert ist. Weitere Informationen finden Sie in der FileMaker Server Hilfe.
2. Öffnen Sie in FileMaker Pro jede FileMaker-Datenbank, die Sie veröffentlichen wollen, und stellen Sie sicher, dass in der Datenbank das erweiterte Zugriffsrecht fmxml für XML Custom Web Publishing aktiviert ist. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 19.

   Um auf XML-Daten in einem Ausschnitt zuzugreifen, stellen Sie die Ansicht für das Datenbanklayout auf Als Formular anzeigen oder Als Liste anzeigen ein. Wenn ein Benutzer oder Script die Ansicht des Datenbanklayouts zu Als Tabelle anzeigen ändert, kann nur auf den ersten Bezugsdatensatz (erste Zeile im Ausschnitt) als XML-Daten zugegriffen werden.

   Die XML-Daten werden in einer Reihenfolge ausgegeben, die der Reihenfolge entspricht, in der die Feldobjekte dem Layout hinzugefügt wurden. Wenn die XML-Datenreihenfolge der Reihenfolge entsprechen soll, in der die Felder auf dem Bildschirm angezeigt werden (von oben nach unten, von links nach rechts), dann wählen Sie alle Felder aus, gruppieren Sie sie und heben Sie dann die Gruppierung wieder auf. Dieser Vorgang setzt die Layoutreihenfolge wieder zurück auf die Bildschirmreihenfolge.
3. Senden Sie durch ein HTML-Formular, einen HREF-Link oder ein Script in Ihrem Programm oder Ihrer Web-Seite eine HTTP- oder HTTPS-Abfrage in Form eines URLs mit Angabe der FileMaker XML-Grammatik, einen Query-Befehl und einen oder mehrere FileMaker-Query-Parameter an die Web Publishing Engine. Der URL kann auch in einem Browser eingegeben werden.

   Informationen über die Angabe des URLs finden Sie im nächsten Abschnitt, „Erläuterung der URL-Syntax für XML-Daten und Medienobjekte.“ Informationen über Query-Befehle und -Parameter finden Sie unter „Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings“ auf Seite 38, und in Anhang A, „Gültige Namen in Query-Strings“.
4. Die Web Publishing Engine verwendet die im URL angegebene Grammatik, um XML-Daten mit den Ergebnissen Ihrer Abfrage zu erzeugen (z. B. eine Gruppe von Datensätzen aus der Datenbank), und gibt diese an Ihr Programm oder Ihren Webbrowser zurück.
5. Der Webbrowser, falls er über einen XML-Parser verfügt, zeigt die Daten an bzw. das Programm verwendet die Daten wie von Ihnen angegeben.

   Wenn Sie ein Client-seitiges Stylesheet angegeben haben, weist der Parser des Webbrowsers auch die Stylesheet-Anweisungen zu. Weitere Informationen finden Sie unter „Verwenden von Server-seitiger und Client-seitiger Stylesheet-Verarbeitung“ auf Seite 41.

Erläuterung der URL-Syntax für XML-Daten und Medienobjekte

Dieser Abschnitt beschreibt die URL-Syntax beim Verwenden der Web Publishing Engine für den Zugriff auf XML-Daten und Medienobjekte in FileMaker-Datenbanken. Die URL-Syntax für die Verwendung von XSLT-Stylesheets unterscheidet sich von XML. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“ auf Seite 52 und „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen“ auf Seite 53.

Erläuterung der URL-Syntax für XML-Daten

Die URL-Syntax beim Verwenden der Web Publishing Engine für den Zugriff auf XML-Daten in FileMaker-Datenbanken lautet:

<scheme>://<host>[:<port>]/fmi/xml/<xml_grammar>.xml[?<query string>]

Dabei gilt:

- <scheme> kann das HTTP- oder HTTPS-Protokoll sein.
- <host> ist die IP-Adresse oder der Domänenname des Hosts, auf dem der Web-Server installiert ist.
- <port> ist optional und gibt den Port an, den der Web-Server überwacht. Ohne Portangabe wird der Standard-Port des Protokolls (Port 80 für HTTP oder Port 443 für HTTPS) verwendet.
- <xml_grammar> ist der Name der FileMaker-XML-Grammatik. Mögliche Werte sind fmresultset.xml, FMPXMLRESULT.xml, FMPXMLLAYOUT.xml oder FMPDSORESULT.xml. Weitere Informationen hierzu finden Sie unter „Verwenden der fmresultset-Grammatik“ auf Seite 32 und „Verwenden anderer FileMaker XML-Grammatiken“ auf Seite 35.
- <query string> ist eine Kombination aus einem Query-Befehl und einem oder mehreren Query-Parametern für FileMaker XML Publishing. (Der Befehl –dbnames verlangt keine Parameter.) Weitere Informationen finden Sie unter „Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings“ auf Seite 38, und Anhang A, „Gültige Namen in Query-Strings“.

Hinweis Die URL-Syntax einschließlich der Namen von Query-Befehl und -Parametern unterscheidet Groß- und Kleinschreibung, mit Ausnahme von Teilen des Query-Strings. Der URL ist vorwiegend in Kleinbuchstaben mit Ausnahme der drei Grammatiknamen in Großbuchstaben: FMPXMLRESULT, FMPXMLLAYOUT und FMPDSORESULT. Informationen über die Regeln für Groß-/Kleinschreibung im Query-String finden Sie unter „Richtlinien für die Verwendung von Query-Befehlen und -Parametern“ auf Seite 88.

Zwei URL-Beispiele für den Zugriff auf XML-Daten über die Web Publishing Engine:

http://server.company.com/fmi/xml/fmresultset.xml?–db=products&–lay=sales&–findall

http://192.168.123.101/fmi/xml/FMPXMLRESULT.xml?–db=products&–lay=sales&–findall

Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XML-Lösungen

In einem generierten XML-Dokument für eine XML-Lösung unterscheidet sich die Syntax für den Verweis auf ein Medienobjekt für Medienfelder, die das tatsächliche Objekt in der Datenbank speichern, und für Medienfelder, die einen Verweis auf das Objekt speichern.

- Wenn ein Medienfeld das tatsächliche Objekt in der Datenbank speichert, verwendet das <data>-Element des Medienfelds die folgende relative URL-Syntax für den Verweis auf das Objekt:

  <data>/fmi/xml/cnt/data.<extension>?<query string></data>

  Dabei gilt: <extension> ist die Dateinamenerweiterung zur Identifizierung des Objekttyps wie beispielsweise .jpg. Die Dateinamenerweiterung stellt den MIME-Typ ein, damit der Webbrowser die Mediendaten korrekt identifizieren kann. Informationen über <query string> finden Sie im vorherigen Abschnitt, „Erläuterung der URL-Syntax für XML-Daten.“

  Beispiel:

  <data>/fmi/xml/cnt/data.jpg?–db=products&–lay=sales&–field=product_image(1)&–recid=2</data>

Hinweis In der generierten XML für ein Medienfeld ist der Wert für den Query-Parameter –field ein vollständig qualifizierter Feldname. Die Zahl in Klammern gibt die Wiederholungsnummer für das Medienfeld an und wird für Wiederhol- und für Einzelfelder generiert. Weitere Informationen finden Sie unter „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 89.

Verwenden Sie zum Abrufen der Mediendaten aus der Datenbank die folgende Syntax:

<scheme>://<host>[:<port>]/fmi/xml/cnt/data.<extension>?<query string>

Informationen über <schema>, <host> oder <port> finden Sie im vorherigen Abschnitt, „Erläuterung der URL-Syntax für XML-Daten.“

Beispiel:

http://www.company.com/fmi/xml/cnt/data.jpg?–db=products&–lay=sales&–field=product_image(1)&–recid=2

- Wenn ein Medienfeld einen Dateiverweis anstelle eines tatsächlichen Objekts speichert, enthält das <data>-Element des Medienfelds einen relativen Pfad, der auf das Objekt verweist. Beispiel:

  <data>/images/logo.jpg</data>

  Hinweis Das Medienobjekt, auf das verwiesen wird, muss im Ordner FileMaker Pro Web gespeichert sein, wenn der Datensatz erstellt oder bearbeitet wird, und dann in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software kopiert oder verschoben werden. Weitere Informationen finden Sie unter „Veröffentlichen des Inhalts von Medienfeldern im Web“ auf Seite 21.
- Für ein leeres Medienfeld ist auch das <data>-Element leer.

Hinweis Die Syntax für Medienobjekte mit XML unterscheidet sich von der Syntax für Medienobjekte mit XSLT. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen“ auf Seite 53.

Erläuterung von URL-Textkodierung

Die URLs für den Zugriff auf XML-Daten und Medienfelder müssen in UTF-8 (Unicode Transformation 8 Bit)-Format kodiert werden. (siehe „Erläuterung von UTF-8-kodierten Daten“ auf Seite 38).

Um z. B. den Wert des Info-Felds in Matinée zu ändern, könnten Sie folgenden URL verwenden:

http://server.company.com/fmi/xml/fmresultset.xml?–db=members&–lay=relationships&–recid=2
&info= matin%C3%A9e&–edit

In diesem Beispiel-URL ist %C3%A9 die URL-kodierte UTF-8-Darstellung des Zeichens é.

Weitere Informationen zur URL-Textkodierung finden Sie in der URL-Spezifikation unter www.w3.org.

Zugriff auf XML-Daten über die Web Publishing Engine

Um über die Web Publishing Engine auf XML-Daten zuzugreifen, verwenden Sie einen URL, der den Namen der zu verwendenden FileMaker-Grammatik, einen FileMaker-Query-Befehl und einen oder mehrere FileMaker-Query-Parameter angibt. Die Web Publishing Engine generiert XML-Daten aus Ihrer Datenbank, die nach einem der folgenden Typen von XML-Grammatiken formatiert werden:

- **fmresultset:** Dies ist die empfohlene Grammatik für die Web Publishing Engine. Sie ist flexibel und für die Erstellung von XSLT-Stylesheets mit bequemerem Feldzugriff nach Name und einfacherer Änderung von relatedset- (Ausschnitt-) Daten optimiert. Diese Grammatik ist auch direkter mit FileMaker-Terminologie verbunden und bietet z. B. globale Speicheroptionen und Identifikation von Statistik- und Formelfeldern. Sie können diese Grammatik für den Zugriff auf XML-Daten und für XSLT-Stylesheets verwenden. Für einfacheres Web Publishing wurde diese Grammatik knapper formuliert als die FMPXMLRESULT-Grammatik. Weitere Informationen finden Sie unter „Verwenden der fmresultset-Grammatik“ auf Seite 32.
- **FMPXMLRESULT und FMPXMLLAYOUT:** Sie können auch die FMPXMLRESULT- und FMPXMLLAYOUT-Grammatiken mit der Web Publishing Engine für den Zugriff auf XML-Daten und für XSLT-Stylesheets verwenden. Um ein einziges Stylesheet sowohl für XML-Export als auch für Custom Web Publishing einzusetzen, müssen Sie die FMPXMLRESULT-Grammatik benutzen. Für den Zugriff auf Wertelisten und Feldanzeigedaten in Layouts müssen Sie die FMPXMLLAYOUT-Grammatik verwenden. Weitere Informationen finden Sie unter „Verwenden anderer FileMaker XML-Grammatiken“ auf Seite 35.
- **FMPDSORESULT:** Die FMPDSORESULT-Grammatik, die in FileMaker Pro für den Export von XML unterstützt wird, steht für den Zugriff auf XML-Daten über die Web Publishing Engine nicht zur Verfügung. Die FMPDSORESULT-Grammatik wird für XSLT-Stylesheets nicht unterstützt. Informationen über die FMPDSORESULT-Grammatik finden Sie in der FileMaker Pro Hilfe.

Abhängig von der Grammatik, die Sie in der URL-Abfrage angegeben haben, generiert die Web Publishing Engine mit einer der folgenden Grammatiken ein XML-Dokument. Jedes XML-Dokument enthält eine XML-Standard-Namespace-Deklaration für die Grammatik. Informationen finden Sie im nächsten Abschnitt, „Erläuterung von Namensräumen für FileMaker XML.“ Verwenden Sie eine dieser Grammatiken in Ihrem Dokument oder auf Ihrer Web-Seite, um FileMaker-Daten im XML-Format anzuzeigen und zu bearbeiten.

Hinweis Von der Web Publishing Engine generierte XML-Daten werden im UTF-8-Format (Unicode Transformation Format 8) kodiert. Weitere Informationen finden Sie unter „Erläuterung von UTF-8-kodierten Daten“ auf Seite 38.

Erläuterung von Namensräumen für FileMaker XML

Mithilfe eindeutiger XML-Namensräume können Sie XML-Tags der richtigen Anwendung zuordnen. Wenn Ihr XML-Dokument beispielsweise zwei <DATABASE>-Elemente enthält, eines für FileMaker Pro-XML-Daten, das andere für Oracle-XML-Daten, kann anhand der Namensräume festgestellt werden, welches <DATABASE>-Element zu welchen XML-Daten gehört.

Die Web Publishing Engine generiert für jede Grammatik einen Standard-Namespace.

Grammatik	Generierter Standard-Namespace
fmresultset	xmlns="http://www.filemaker.com/xml/fmresultset"
FMPXMLRESULT	xmlns="http://www.filemaker.com/ fmpxmlresult"
FMPXMLLAYOUT	xmlns="http://www.filemaker.com/fmpxmllayout"

Erläuterung der FileMaker-Datenbankfehlercodes

Die Web Publishing Engine gibt in den „error code“-Elementen am Beginn jedes XML-Dokuments einen Fehlercode für einen etwaigen Fehler zurück, der im zuletzt ausgeführten Query-Befehl aufgetreten ist. Null (0) bedeutet keinen Fehler.

Grammatik	Verwendete Syntax
fmresultset	<error code="0"></error>
FMPXMLRESULT	<ERRORCODE>0</ERRORCODE>
FMPDSORESULT	<ERRORCODE>0</ERRORCODE>

Das „error code“-Element im XML-Dokument gibt Fehler an, die die Datenbank und Query-Strings betreffen. Andere Fehlertypen können für XSLT-Stylesheets auftreten und werden anders behandelt. Weitere Informationen finden Sie unter Anhang B, „Fehlercodes für Custom Web Publishing“.

Abrufen der Document Type Definitions für die FileMaker-Grammatiken

Sie können die Document Type Definitions (DTDs) für die FileMaker-Grammatiken mithilfe einer HTTP-Abfrage abrufen.

Grammatik	HTTP-Abfrage
fmresultset	http://<host>[:<port]/fmi/xml/fmresultset.dtd
FMPXMLRESULT	http://<host>[:<port]/fmi/xml/FMPXMLRESULT.dtd
FMPXMLLAYOUT	http://<Host>[:<Port]/fmi/xml/FMPXMLLAYOUT.dtd
FMPDSORESULT	http://<host>[:<port]/fmi/xml/FMPDSORESULT.dtd?–db=<database>&–lay=<layout>

Verwenden der fmresultset-Grammatik

Die XML-Elementnamen in dieser Grammatik verwenden FileMaker-Terminologie und die Speicherung von Feldern erfolgt getrennt von den Feldtypen. Die Grammatik umfasst auch die Fähigkeit, Statistik-, Formel- und Variablenfelder zu identifizieren.

Geben Sie zur Verwendung der fmresultset-Grammatik den folgenden Namen der fmresultset-Grammatik im URL an, der das XML-Dokument von der Web Publishing Engine anfordert:

fmresultset.xml

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–findall

Hinweis Geben Sie die fmresultset-Grammatik unbedingt in Kleinbuchstaben an.

Die Web Publishing Engine generiert ein XML-Dokument mithilfe der fmresultset-Grammatik. Im XML-Dokument verweist die Web Publishing Engine auf die Document Type Definition für die fmresultset-Grammatik in der Anweisung <!DOCTYPE> in der zweiten Zeile des Dokuments, unmittelbar nach der Anweisung <?xml...?>. Die Anweisung <!DOCTYPE> gibt den URL für das Herunterladen der DTD für die fmresultset-Grammatik an.

Beschreibung der Elemente der fmresultset-Grammatik

Die fmresultset-Grammatik besteht hauptsächlich aus dem Element <datasource>, dem Element <metadata> und dem Element <resultset>.

Element <datasource>

In der fmresultset-Grammatik enthält das <datasource>-Element die Attribute „table“, „layout“, „date-format“, „time-format“, „timestamp-format“, „total-count“ und „database“.

- Das Attribut „date-format“ des <datasource>-Elements gibt das Format von Datumsangaben im XML-Dokument an:

  dd.MM.yyyy

Dabei gilt:

  - MM ist der zweistellige Wert für den Monat (01 bis 12, wobei 01 für Januar und 12 für Dezember steht).
  - dd ist der zweistellige Wert für den Tag des Monats (00 bis 31).
  - yyyy ist der vierstellige Wert für das Jahr.

- Das Attribut „time-format“ des <datasource>-Elements gibt das Format von Zeitangaben im XML-Dokument an.

  HH:mm:ss

Dabei gilt:

  - HH ist der zweistellige Wert für Stunden (00 bis 23 im 24-Stunden-Format).
  - mm ist der zweistellige Wert für Minuten (00 bis 59).
  - ss ist der zweistellige Wert für Sekunden (00 bis 59).

- Das Attribut „timestamp-format“ des <datasource>-Elements gibt das Format von Datums- und Zeitangaben in einem Zeitstempel an.

  dd.MM.yyyy HH:mm:ss

Element <metadata>

Das <metadata>-Element der fmresultset-Grammatik enthält eines oder mehrere <field-definition>- und <relatedset-definition>-Elemente, jeweils mit Attributen für eines der Felder in der Ergebnismenge.

Die Attribute <field-definition> geben an:

- ob das Feld ein Feld mit automatischer Eingabe ist („yes“ oder „no“)
- ob das Feld ein Feld mit vierstelligem Jahr ist („yes“ oder „no“)
- ob das Feld ein globales Feld ist („yes“ oder „no“)
- die maximale Anzahl an Wiederholwerten (Attribut max-repeat)
- die maximale Anzahl an zulässigen Zeichen (Attribut max-characters)
- ob das Feld ein nicht leeres Feld ist („yes“ oder „no“)
- ob das Feld ein Feld nur mit numerischen Daten ist („yes“ oder „no“)
- result („text“, „number“, „date“, „time“, „timestamp“ oder „container“)
- ob das Feld ein Uhrzeit-des-Tages-Feld ist („yes“ oder „no“)
- type („normal“, „calculation“ oder „summary“)
- den Feldnamen (bei Bedarf vollständig qualifiziert)

Das <relatedset-definition>-Element repräsentiert einen Ausschnitt. Jedes Bezugsfeld in einem Ausschnitt wird durch das <field-definition>-Element innerhalb des <relatedset-definition>-Elements dargestellt. Wenn ein Ausschnitt mehrere Bezugsfelder enthält, werden die Felddefinitionen für die Bezugsfelder in einem einzigen <relatedset-definition>-Element gruppiert.

Element <resultset>

Das <resultset>-Element enthält alle <record>-Elemente, die als Ergebnis einer Query zurückgegeben wurden, sowie ein Attribut für die Gesamtzahl der gefundenen Datensätze. Jedes <record>-Element enthält die Felddaten für einen Datensatz in der Ergebnismenge – einschließlich der Attribute mod-id und record-id für den Datensatz – sowie das <data>-Element mit den Daten für ein Feld im Datensatz.

Jeder Datensatz in einem Ausschnitt wird durch ein <record>-Element innerhalb des <relatedset>-Elements dargestellt. Das Attribut „count“ des <relatedset>-Elements gibt die Anzahl der Datensätze in einem Ausschnitt an und das Attribut „table“ gibt die mit dem Ausschnitt verbundene Tabelle an.

Beispiel für XML-Daten in der fmresultset-Grammatik

Nachfolgend erhalten Sie ein Beispiel für XML-Daten, die mit der fmresultset-Grammatik generiert wurden.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE fmresultset PUBLIC "-//FMI//DTD fmresultset//EN" ""http://localhost:16014/fmi/xml/fmresultset.dtd">
 <fmresultset xmlns="http://www.filemaker.com/xml/fmresultset" version="1.0">
 <error code="0" />
 <product build="07.02.2007" name="FileMaker Web Publishing Engine" version="9.0.1.36" />
 <datasource database="art" date-format="MM/dd/yyyy" layout="web3" table="art" time-format="HH:mm:ss" timestamp-format="MM/dd/yyyy HH:mm:ss" total-count="12" />
 <metadata>
 <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="Title" not-empty="no" numeric-only="no" result="text" time-of-day="no" type="normal" />
 <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="Artist" not-empty="no" numeric-only="no" result="text" time-of-day="no" type="normal" />
 <relatedset-definition table="artlocations">
 <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="artlocations::Location" not-empty="no" numeric-only="no" result="text" time-of-day="no" type="normal" />
 <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="artlocations::Date" not-empty="no" numeric-only="no" result="date" time-of-day="no" type="normal" />
 </relatedset-definition>
 <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="Style" not-empty="no" numeric-only="no" result="text" time-of-day="no" type="normal" />
 <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="Length" not-empty="no" numeric-only="no" result="number" time-of-day="no" type="calculation" />
 </metadata>
 <resultset count="1" fetch-size="1">
 <record mod-id="6" record-id="14">
 <field name="Title">
 <data>Spring in Giverny 3</data>
 </field>
 <field name="Artist">
 <data>Claude Monet</data>
 </field>
 <relatedset count="0" table="artlocations" />
 <field name="Style">
 <data />
 </field>
 <field name="length">
 <data>19</data>
 </field>
 </record>
 </resultset>
 </fmresultset>
```

Verwenden anderer FileMaker XML-Grammatiken

Die anderen FileMaker XML-Grammatiken enthalten Informationen über Feldtypen, Wertelisten und Layouts. FMPXMLRESULT und fmresultset sind hinsichtlich der Funktionalität äquivalent. Für den Zugriff auf Wertelisten und Feldanzeigedaten in Layouts müssen Sie die FMPXMLLAYOUT-Grammatik verwenden. Die FMPXMLRESULT- und FMPXMLLAYOUT-Grammatiken sind kompakter für den Datenaustausch.

Geben Sie zur Verwendung der FMPXMLRESULT-Grammatik den folgenden Grammatiknamen im URL an, der das XML-Dokument von der Web Publishing Engine anfordert:

FMPXMLRESULT.xml

Beispiel:

http://192.168.123.101/fmi/xml/FMPXMLRESULT.xml?–db=employees&–lay=family&–findall

Geben Sie zur Verwendung der FMPXMLLAYOUT-Grammatik den folgenden Grammatiknamen mit dem Query-Befehl -view im URL an, der das XML-Dokument von der Web Publishing Engine anfordert:

FMPXMLLAYOUT.xml

Beispiel:

http://192.168.123.101/fmi/xml/FMPXMLLAYOUT.xml?–db=employees&–lay=family&–view

Hinweis Geben Sie die FMPXMLRESULT- und die FMPXMLLAYOUT-Grammatik unbedingt in Großbuchstaben an.

Im generierten XML-Dokument verweist die Web Publishing Engine auf die Document Type Definition für die fmresultset-Grammatik in der Anweisung <!DOCTYPE> in der zweiten Zeile des Dokuments, unmittelbar nach der Anweisung <?xml...?>. Die Anweisung <!DOCTYPE> gibt den URL für das Herunterladen der DTD für die Grammatik an.

Beschreibung der Elemente in der FMPXMLRESULT-Grammatik

In der FMPXMLRESULT-Grammatik enthält das <DATABASE>-Element die Attribute NAME, RECORDS, DATEFORMAT, LAYOUT und TIMEFORMAT.

Das Attribut DATEFORMAT des <DATABASE>-Elements gibt das Datumsformat im XML-Dokument an. Das Attribut TIMEFORMAT des <DATABASE>-Elements gibt das Zeitformat im XML-Dokument an. Datums- und Zeitformate der Grammatiken FMPXMLRESULT und fmresultset sind gleich (siehe dazu die Tabellen unter „Beschreibung der Elemente der fmresultset-Grammatik“ auf Seite 32.)

Das <METADATA>-Element der FMPXMLRESULT-Grammatik enthält eines oder mehrere <FIELD>-Elemente. Jedes FIELD-Element enthält Daten für ein Feld bzw. eine Spalte in der Ergebnismenge einschließlich des Feldnamens, wie er in der Datenbank definiert ist , den Feldtyp, die Ja/Nein-Zulässigkeit für leere Felder (Attribut EMPTYOK) und die maximale Anzahl von Wiederholwerten (Attribut MAXREPEAT). Gültige Werte für Feldtypen sind TEXT, NUMBER, DATE, TIME, TIMESTAMP und CONTAINER.

Das <RESULTSET>-Element enthält alle <ROW>-Elemente, die als Ergebnis einer Query zurückgegeben wurden, sowie ein Attribut für die Gesamtzahl der gefundenen Datensätze. Jedes <ROW>-Element enthält die Feld-/Spalten-Daten für jede Zeile in der Ergebnismenge. Diese Daten umfassen RECORDID und MODID für die Zeile (siehe „Query-Parameter –modid (Änderungs-ID)“ auf Seite 98) und das <COL>-Element. Das <COL>-Element enthält die Daten für ein Feld/eine Spalte in der Zeile, wobei mehrere <DATA>-Elemente einen der Werte in einem Wiederhol- oder Ausschnittfeld repräsentieren.

Beispiel für XML-Daten in der FMPXMLRESULT-Grammatik

Nachfolgend sehen Sie ein Beispiel für XML-Daten, die mit der FMPXMLRESULT-Grammatik generiert wurden.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE FMPXMLRESULT PUBLIC "-//FMI//DTD FMPXMLRESULT//EN" ""http://localhost:16014/fmi/xml/
FMPXMLRESULT.dtd">
<FMPXMLRESULT xmlns="http://www.filemaker.com/fmpxmlresult">
 <ERRORCODE>0</ERRORCODE>
 <PRODUCT BUILD="07.02.2007" NAME="FileMaker Web Publishing Engine" VERSION="9.0.1.36" />
 <DATABASE DATEFORMAT="MM/dd/yyyy" LAYOUT="web" NAME="art" RECORDS="12" TIMEFORMAT="HH:mm:ss" />
 <METADATA>
  <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Title" TYPE="TEXT" />
   <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Artist" TYPE="TEXT" />
  <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Image" TYPE="CONTAINER" />
 </METADATA>
 <RESULTSET FOUND="1">
 <ROW MODID="6" RECORDID="15">
 <COL>
  <DATA>Spring in Giverny 4</DATA>
 </COL>
 <COL>
  <DATA>Claude Monet</DATA>
 </COL>
 <COL>
  <DATA>/fmi/xml/cnt/data.jpg?-db=art&-lay=web&-recid=15&-field=Image(1)</DATA>
 </COL>
 </ROW>
 </RESULTSET>
</FMPXMLRESULT>
```

Die Reihenfolge der <COL>-Elemente entspricht der Reihenfolge der <FIELD>-Elemente im <METADATA>-Element – z. B. entspricht die Reihenfolge der Felder „Title“ und „Artist“ im <METADATA>-Element der Reihenfolge von „Village Market“ und „Camille Pissarro“ in den Elementen <RESULTSET> und <ROW>.

Beschreibung der Elemente der Grammatik FMPXMLLAYOUT

In der FMPXMLLAYOUT-Grammatik enthält das <LAYOUT>-Element den Namen des Layouts, den Namen der Datenbank sowie <FIELD>-Elemente für jedes Feld im entsprechenden Layout der Datenbank. Jedes <FIELD>-Element beschreibt den Feldtyp und enthält das Attribut VALUELIST für jede Werteliste des Feldes.

Das <VALUELISTS>-Element enthält eines oder mehrere <VALUELIST>-Elemente für jede Werteliste im Layout – jedes VALUELIST-Element enthält den Namen der Werteliste und ein <VALUE>-Element für jeden Wert in der Liste.

Für Datums-, Zeit- und Zeitstempelfelder werden Daten für Wertelisten über das „fm“-Format für diesen Feldtyp formatiert. Die „fm“-Formate sind MM/tt/jjjj für Datum, HH:mm:ss für Zeit und MM/tt/jjjj HH:mm:ss für Zeitstempel. Weitere Informationen finden Sie unter „Verwenden der Datums-, Zeit- und Tages-Erweiterungsfunktionen“ auf Seite 72. Wenn zum Beispiel eine Werteliste „Geburtstage“ für ein Einblendmenü für ein Feld „Geburtstag“ eines Layouts verwendet wird und das Feld „Geburtstag“ ein Datumsfeld ist, werden alle für diese Werteliste ausgegebenen Werte im Datumsformat „fm“ ausgegeben.

Hinweis Wenn zwei Felder mit unterschiedlichem Feldtyp in einem Layout die gleiche Werteliste verwenden, bestimmt das erste Feld das Format der Daten der Werteliste.

XML-Daten in der FMPXMLLAYOUT-Grammatik – Beispiel

Im Folgenden finden Sie ein Beispiel für XML-Daten, die mit der FMPXMLLAYOUT-Grammatik generiert wurden.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE FMPXMLLAYOUT PUBLIC "-//FMI//DTD FMPXMLLAYOUT//EN" ""http://localhost:16014/fmi/xml/
FMPXMLLAYOUT.dtd">
 <FMPXMLLAYOUT xmlns="http://www.filemaker.com/fmpxmllayout">
 <ERRORCODE>0</ERRORCODE>
 <PRODUCT BUILD="07.02.2007" NAME="FileMaker Web Publishing Engine" VERSION="9.0.1.36" />
 <LAYOUT DATABASE="art" NAME="web2">
 <FIELD NAME="Title">
  <STYLE TYPE="EDITTEXT" VALUELIST="" />
 </FIELD>
 <FIELD NAME="Artist">
  <STYLE TYPE="EDITTEXT" VALUELIST="" />
 </FIELD>
 <FIELD NAME="Image">
  <STYLE TYPE="EDITTEXT" VALUELIST="" />
 </FIELD>
 <FIELD NAME="artlocations::Location">
  <STYLE TYPE="EDITTEXT" VALUELIST="" />
 </FIELD>
 <FIELD NAME="artlocations::Date">
  <STYLE TYPE="EDITTEXT" VALUELIST="" />
 </FIELD>
 <FIELD NAME="Style">
  <STYLE TYPE="POPUPMENU" VALUELIST="style"/>
 </FIELD>
 </LAYOUT>
  <VALUELISTS>
   <VALUELIST NAME="style">
    <VALUE>Impressionist     </VALUE>
    <VALUE>Modern     </VALUE>
    <VALUE>Abstract     </VALUE>
   </VALUELIST>
  </VALUELISTS>
 </FMPXMLLAYOUT>
```

Erläuterung von UTF-8-kodierten Daten

Alle von der Web Publishing Engine generierten XML-Daten werden im UTF-8-Format (Unicode Transformation Format 8) kodiert. Dieses Format komprimiert ASCII-Zeichen im Unicode-Standardformat von 16 Bit auf 8 Bit. Für Unicode und UFT-8 sind XML-Parser erforderlich.

UTF-8-Kodierung unterstützt die direkte Darstellung der Werte 0-127 für den ASCII-Standardzeichensatz in der englischen Sprache und bietet Multibyte-Kodierungen für Unicode-Zeichen mit höheren Werten.

Hinweis Stellen Sie sicher, dass Sie einen Webbrowser oder ein Textverarbeitungsprogramm verwenden, das UTF-8-Dateien unterstützt.

Das Kodierungsformat UTF-8 hat die folgenden Eigenschaften:

- Alle ASCII-Zeichen sind UTF-8-Zeichen von der Größe eines Byte. Eine gültige ASCII-Zeichenkette ist eine gültige UTF-8-Zeichenkette.
- Jedes Nicht-ASCII-Zeichen (beliebiges Zeichen mit gesetztem höherwertigem Bit) ist Teil eines Multibyte-Zeichens.
- Das erste Byte eines UTF-8-Zeichens gibt die Anzahl der zusätzlichen Bytes im Zeichen an.
- Das erste Byte eines Multibyte-Zeichens lässt sich leicht vom darauf folgenden Byte unterscheiden, wodurch sich der Beginn eines Zeichens von einer beliebigen Position in einem Datenstrom einfach ermitteln lässt.
- Eine Konvertierung zwischen UTF-8 und Unicode ist einfach vorzunehmen.
- Die UTF-8-Kodierung ist relativ kompakt. Bei Text, der einen hohen Prozentsatz an ASCII-Zeichen enthält, ist sie kompakter als Unicode. Im schlimmsten Fall ist ein UTF-8-String nur 50% größer als der entsprechende Unicode-String.

Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings

Um XML-Daten aus einer FileMaker-Datenbank abzufragen, verwenden Sie die FileMaker-Query-Befehle und -Parameter in einem Query-String. Sie können beispielsweise den Query-Befehl –findall im folgenden Query-String in einem URL verwenden, um eine Liste aller Produkte aus einer FileMaker-Datenbank mit dem Namen products anzufordern:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=products–lay=sales&–findall

Ein Query-String darf nur einen Query-Befehl, z. B. –new, enthalten. Die meisten Query-Befehle verlangen auch mehrere entsprechende Query-Parameter im Query-String. Beispielsweise verlangen alle Query-Befehle außer –dbnames den Parameter –db, der die abzufragende Datenbank angibt.

Sie können auch Query-Befehle und -Parameter in einem URL oder in einer <?xslt–cwp–query?>-Verarbeitungsanweisung in einem FileMaker XSLT-Stylesheet verwenden. Weitere Informationen finden Sie unter Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Dieser Abschnitt bietet eine Zusammenfassung der FileMaker Query-Befehle und -Parameter. Weitere Informationen über ihre Verwendung in einem Query-String finden Sie in Anhang A, „Gültige Namen in Query-Strings“.

Hinweis Die Web Publishing Engine unterstützt auch einen zusätzlichen Query-Befehl (-process) sowie drei Query-Parameter, die nur zur Verwendung mit FileMaker XSLT-Stylesheets definiert sind. Weitere Informationen finden Sie unter „Verwenden von Query-Strings in FileMaker XSLT-Stylesheets“ auf Seite 54.

Verwenden Sie diesen Query-Befehlsnamen	Zur Ausführung dieses Befehls
–dbnames	Namen aller bereitgestellten und im Web gemeinsam genutzten Datenbanken abrufen.
–delete	Datensatz löschen.
–dup	Datensatz duplizieren.
–edit	Datensatz bearbeiten.
–find	Datensatz/Datensätze suchen.
–findall	Nach allen Datensätzen suchen.
–findany	Zufällig ermittelten Datensatz anzeigen.
–findquery	Komplexe oder zusammengesetzte Suchabfrage durchführen.
–layoutnames	Namen aller verfügbaren Layouts für eine bereitgestellte und im Web gemeinsam genutzte Datenbank abrufen.
–new	Neuen Datensatz hinzufügen.
–scriptnames	Namen aller verfügbaren Scripts für eine bereitgestellte und im Web gemeinsam genutzte Datenbank abrufen.
–view	Layout-Informationen aus einer Datenbank abrufen, wenn die FMPXMLLAYOUT-Grammatik angegeben ist. Ruft den Abschnitt <metadata> des XML-Dokuments und eine leere Datensatzmenge ab, wenn die fmresultset- oder FMPXMLRESULT-Grammatik angegeben ist.

Verwenden Sie diese Query-Parameternamen	Mit diesen Query-Befehlen
–db (Datenbankname)	Erforderlich für alle Query-Befehle mit Ausnahme von –dbnames und –process (nur XSLT-Abfragen)
–delete.related	Optional bei –edit
–field	Erforderlich zur Angabe eines Felds in einem URL für Medienabfragen. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XML-Lösungen“ auf Seite 28.
Feldname	Mindestens ein Feldname ist für –edit erforderlich. Optional bei –find. Weitere Informationen finden Sie unter „Query-Parameter „fieldname“ (Nicht-Medienfeldname)“ auf Seite 96.
Feldname.op (Operator)	Optional bei –find
–lay (Layoutname)	Erforderlich für alle Query-Befehle mit Ausnahme von –dbnames, –layoutnames, –scriptnames und –process (nur XSLT-Abfragen)
–lay.response (Layout für XML-Antwort wechseln)	Optional für alle Query-Befehle mit Ausnahme von –dbnames, –layoutnames, –scriptnames und –process (nur XSLT-Abfragen)
–lop (logischer Operator)	Optional bei –find
–max (maximale Datensätze)	Optional bei –find, –findall
–modid (Änderungs-ID)	Optional bei –edit

Verwenden Sie diese Query-Parameternamen	Mit diesen Query-Befehlen
–query	Erforderlich bei zusammengesetzten Suchabfragen -findquery.
–recid (Datensatz-ID)	Erforderlich bei –edit, –delete, –dup. Optional bei –find
–relatedsets.filter	Optional bei –find, –edit, –new, –dup und –findquery.
–relatedsets.max	Optional bei –find, –edit, –new, –dup und –findquery.
–script (Script ausführen)	Optional bei –find, –findall, –findany, –new, –edit, –delete, –dup, –view
–script.param (einen Parameterwert an das durch –script angegebene Script übergeben)	Optional bei –script
–script.prefind (Script vor –find, –findany und –findall ausführen)	Optional bei –find, –findany, –findall
–script.prefind.param (einen Parameterwert an das durch –script.prefind angegebene Script übergeben)	Optional bei –script.prefind
–script.presort (Script vor Sortierung ausführen)	Optional bei –find, –findall
–script.presort.param (einen Parameterwert an das durch –script.presort angegebene Script übergeben)	Optional bei –script.presort
–skip (Datensätze überspringen)	Optional bei –find, –findall
–sortfield.[1-9] (Sortierfeld)	Optional bei –find, –findall
–sortorder.[1-9] (Sortierfolge)	Optional bei –find, –findall
–stylehref (Stylesheet HREF)	Optional für alle Query-Befehle (zur Angabe eines Stylesheet-URLs für –styletype)
–styletype (Stylesheet-Typ)	Optional für alle Query-Befehle (zur Angabe eines Client-seitigen Stylesheets)

Wechseln des Layouts für eine XML-Antwort

Der Query-Parameter –lay gibt das Layout an, das Sie beim Abfragen von XML-Daten verwenden wollen. Häufig eignet sich dasselbe Layout für die Verarbeitung der Daten, die von der Abfrage geliefert werden. In einigen Fällen sollten Sie die Daten in einem Layout durchsuchen, das Felder enthält, die aus Sicherheitsgründen in keinem Layout vorhanden sind, mit dem die Ergebnisse angezeigt werden. (Um Daten in einem Feld zu suchen, muss das Feld auf das Layout gesetzt werden, das Sie in der XML-Abfrage angeben.)

Um für die Anzeige der XML-Antwort ein anderes Layout als für die Verarbeitung der XML-Abfrage zu verwenden, können Sie den optionalen Query-Parameter –lay.response angeben.

Die folgende Abfrage sucht z. B. Werte größer 100,000 im Feld Salary im Layout Budget. Die gefundenen Daten werden im Layout ExecList angezeigt, das das Feld Salary nicht enthält.

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=Budget&Salary=100000&Salary.op=gt&–find&–lay.response=VorstListe

Erläuterung der Verarbeitung einer XML-Abfrage

Es gibt mehrere Query-Parameter, die die Verarbeitung einer XML-Abfrage und die Generierung eines XML-Dokuments beeinflussen.

FileMaker Server und die Web Publishing Engine verarbeiten eine XML-Abfrage in dieser Reihenfolge:

1. Verarbeitung des –lay -Query-Parameters
2. Festlegen der globalen Feldwerte, die in der Query angegeben sind (der Teil .global= des URLs)
3. Verarbeitung des Query-Parameters –script.prefind, falls angegeben
4. Verarbeitung der Query-Befehle wie z. B. –find oder –new
5. Verarbeitung des Query-Parameters –script.presort, falls angegeben
6. Sortieren der Ergebnisdaten, falls eine Sortierung angegeben wurde
7. Verarbeitung des Query-Parameters –lay.response, um das Layout zu wechseln, falls angegeben
8. Verarbeitung des Query-Parameters –script, falls angegeben
9. Generieren des XML-Dokuments.

Wenn einer der obigen Schritte einen Fehlercode erzeugt, stoppt die Abfrageverarbeitung und etwaige nachfolgende Schritte werden nicht ausgeführt. Jedoch bleiben alle vorangehenden Schritte in der Abfrage ausgeführt.

Betrachten Sie z. B. eine Abfrage, die den aktuellen Datensatz löscht, die Datensätze sortiert und dann ein Script ausführt. Wenn der Parameter –sortfield ein nicht vorhandenes Feld angibt, löscht die Abfrage den aktuellen Datensatz und gibt dann Fehlercode 102 (Feld fehlt) zurück, führt aber das Script nicht aus.

Verwenden von Server-seitiger und Client-seitiger Stylesheet-Verarbeitung

Die Web Publishing Engine unterstützt Server-seitige Verarbeitung eines XSLT-Stylesheets und gestattet Ihnen auch die Verwendung eines Query-Parameters, der Client-seitige Stylesheet-Verarbeitung angibt.

Es ist wichtig, die Unterschiede zwischen den beiden Arten der Stylesheet-Verarbeitung sowie die Implikationen für die Sicherheit bei Client-seitiger Verarbeitung zu verstehen. Server-seitige Verarbeitung gibt Web-Benutzern keinen Zugang zu den ungefilterten XML-Daten und ist daher sicherer als Client-seitige Verarbeitung. Bei Server-seitiger Verarbeitung werden die Daten in einer Form präsentiert, die der Eigentümer der Daten oder der Verfasser des XSLT-Stylesheets für angemessen hält. Server-seitige Verarbeitung verbirgt die Datenbanknamen, Feldnamen und andere Implementierungsdetails vor Web-Benutzern. Server-seitige Verarbeitung kann auch für die Angabe statisch definierter Query-Parameter benutzt werden, die die Verwendung nicht zugelassener Query-Befehle und Query-Parameter verhindert, z. B. Datenbanknamen. Weitere Informationen finden Sie unter Kapitel 5, „Einführung in Custom Web Publishing mit XSLT“ und Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Wenn Ihre Lösung Client-seitige Stylesheet-Verarbeitung erfordert, können Sie über die Web Publishing Engine eine Anweisung für die XML-Stylesheet-Verarbeitung mit jeder Grammatik generieren, indem Sie die Parameter –styletype und –stylehref in die FileMaker-Query-Stringabfrage einfügen. Sie können Cascading Style Sheets (CSS) oder XSLT-Stylesheets für die Anzeige Ihres XML-Dokuments verwenden.

- Mit dem Parameter –styletype wird der Wert des type-Attributs festgelegt (type=text/css oder type=text/xsl).
- Mit dem Parameter –stylehref wird der Wert des HREF-Attributs festgelegt, das die Position des Stylesheets mit einem absoluten Pfad angibt. Beispiel: href=/mystylesheet.css oder href=/stylesheets/mystylesheet.xsl Das Stylesheet kann einen beliebigen Namen haben, aber die Erweiterung muss .css oder .xsl sein.

Beispiel für einen FileMaker-Query-String, der Client-seitige Stylesheet-Verarbeitung erzeugt:

http://localhost/fmi/xml/fmresultset.xml?–db=products–lay=sales&–findall&–styletype=text/xsl&–stylehref=/mystylesheet.xsl

Hinweis Der Schrägstrich / in –stylehref=/document.xsl in diesem Beispiel wird verwendet, da das Stylesheet sich im Root-Ordner der Web-Server-Software befindet. Verwenden Sie für das Stylesheet einen URL mit einem absoluten Pfad, um die Position auf dem Web-Server anzugeben. Das Stylesheet kann sich auch auf einem anderen Web-Server befinden.

Auf der Grundlage dieser Abfrage nimmt die Web Publishing Engine die folgende Verarbeitungsanweisung in das XML-Dokument auf:

<?xml–stylesheet type="text/xsl" href="/mystylesheet.xsl"?>

Kopieren oder platzieren Sie das Stylesheet für Client-seitige Verarbeitung auf den Web-Server an die Stelle, die der absolute Pfad im URL für das HREF-Attribut angibt.

Wichtig Platzieren Sie Stylesheets für Client-seitige Verarbeitung nicht in den Ordner xslt-template-files, der für Server-seitige XSLT-Stylesheet-Verarbeitung verwendet wird. Weitere Informationen finden Sie unter „Verwenden von FileMaker XSLT-Stylesheets in einer Website oder einem Programm“ auf Seite 48.

Hinweis Einige Webbrowser unterstützen keine Client-seitige Verarbeitung. Weitere Informationen finden Sie in der Dokumentation zu Ihrem Webbrowser.

Fehlerbehebung beim XML-Dokumentzugriff

Wenn beim Zugriff auf XML-Dokumente mit der Web Publishing Engine Probleme auftreten, prüfen Sie Folgendes:

- Die erweiterten Zugriffsrechte in der Datenbank müssen auf „XML Custom Web Publishing“ eingestellt und einem Benutzerkonto zugewiesen sein. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 19.
- Die Datenbank wird auf der Datenbank-Server-Komponente des FileMaker Server-Einsatzes bereitgestellt und von FileMaker Server geöffnet. Weitere Informationen finden Sie in der FileMaker Server Hilfe.
- Prüfen Sie, ob der verwendete Datenbankkontoname und ggf. Ihr Passwort korrekt sind.
- Die Web-Server-Komponente des FileMaker Server-Einsatzes wird ausgeführt.
- Die Web Publishing Engine-Komponente des FileMaker Server-Einsatzes wird ausgeführt.
- XML Publishing ist in der Web Publishing Engine-Komponente aktiviert. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

Kapitel 5
Einführung in Custom Web Publishing mit XSLT

FileMaker XSLT-Stylesheets ermöglichen das Umwandeln, Filtern oder Formatieren von XML-Daten für die Verwendung in einem Webbrowser oder in anderen Programmen und Anwendungen. Dieses Kapitel stellt FileMaker XSLT-Stylesheets und ein Tool vor, das Ihnen den Einstieg in die Erstellung von XSLT-Stylesheets erleichtert: den FileMaker XSLT-Site-Assistenten. Weitere Informationen über den Aufbau von FileMaker XSLT-Stylesheets finden Sie in Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Erläuterung von FileMaker XSLT-Stylesheets

Mithilfe von FileMaker XSLT-Stylesheets können Sie:

- FileMaker-Daten filtern, indem Sie steuern, welche Datenbankfelder vom Stylesheet veröffentlicht werden.
- Metadaten wie z. B. Datenbank- und Feldnamen ausblenden.
- das Erscheinungsbild der Daten in einer Web-Seite formatieren und steuern, wie der Web-Benutzer mit den Daten interagiert.
- die Daten als HTML oder Text ausgeben, z. B. vCards oder durch Komma getrennte Werte.
- die Daten von einer FileMaker XML-Grammatik in eine andere XML-Grammatik umwandeln, um sie in einer anderen Datenbank oder Anwendung zu nutzen, z. B. Scalable Vector Graphics (SVG).
- eine Untergruppe der FileMaker-Daten in andere Websites und mit anderen Middleware- und Spezialanwendungen integrieren, die sich erheblich von der FileMaker-Datenbank unterscheiden können.
- die veröffentlichten Feldnamen ändern und unbefugte Verwendung der Datenbankdesigninformation verhindern.

Hinweis Custom Web Publishing mit XSLT für FileMaker Server basiert auf der W3C-Empfehlung für XSLT 1.0. Informationen über XSLT 1.0 finden Sie unter www.w3.org. Zusätzliche Funktionalität wie Sitzungsverwaltung, E-Mail-Versand und Zugriff auf Cookies und Header wird durch FileMaker XSLT-Erweiterungsfunktionen zur Verfügung gestellt. Weitere Informationen finden Sie unter „Verwenden der FileMaker XSLT-Erweiterungsfunktionen und -parameter“ auf Seite 59. Die Web Publishing Engine unterstützt nicht XSL Formatting Objects (XSL-FO).

Beispiele für die Verwendung von FileMaker XSLT-Stylesheets

Ein paar Beispiele für die unzähligen Möglichkeiten mit FileMaker XSLT-Stylesheets:

- Sie können eine Tabelle in eine Web-Seite mit einem Teil der Daten aus einer FileMaker-Datenbank einfügen, damit Web-Benutzer darin blättern können. Die Tabelle könnte z. B. Namen und Adressen, aber keine Telefonnummern enthalten. Um unbefugten Zugriff zu verhindern, kann die Web-Seite statt der tatsächlichen Feldnamen in der FileMaker-Datenbank (z. B. „vor_name“) generische Beschriftungen für die Daten zeigen (z. B. „Name“).
- Sie können eine Web-Seite oder eine Anwendung erstellen, die Daten aus einem FileMaker-Ausschnitt in Informationen aus anderen Datenquellen integriert.

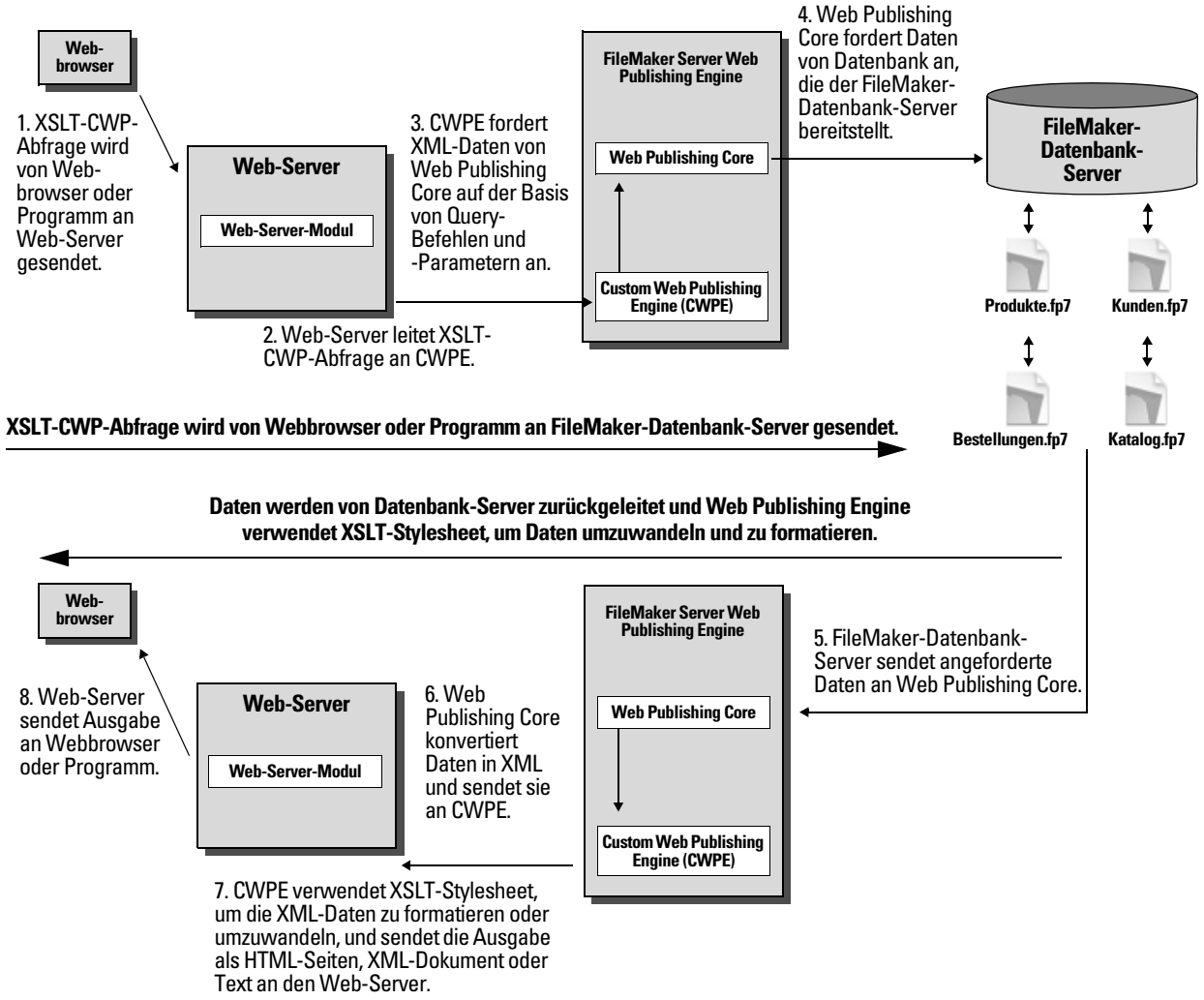
- Sie können einer Web-Seite eine Taste hinzufügen, über die eine vCard aus den Kontaktinformationen einer Person in einer FileMaker-Datenbank erzeugt wird.
- Sie können die XML-Daten aus einer FileMaker-Datenbank in eine XML-Grammatik umwandeln, die eine Tabellenkalkulations- oder Datenbankanwendung öffnen kann.

Einstieg in die Verwendung von Custom Web Publishing mit XSLT

Wenn Sie Standard-XML und XSLT kennen, können Sie sofort mit dem Einsatz der Web Publishing Engine beginnen, nachdem Sie ein paar spezielle Details über FileMaker XML und XSLT Publishing wie die Verwendung von FileMaker-XSLT-Erweiterungsfunktionen, Query-Befehlen und Query-Parametern gelernt haben. Der XSLT-Site-Assistent hilft Ihnen bei der Erstellung von Stylesheets und zeigt Ihnen den Aufbau der Stylesheets. Sie können die Stylesheets mithilfe Ihres bevorzugten XML- und XSLT-Erstellungswerkzeugs weiter ausarbeiten.

Wie die Web Publishing Engine auf der Basis von XML-Daten und XSLT-Stylesheets Seiten generiert

Nachdem eine XSLT Custom Web Publishing (XSLT-CWP)-Abfrage an den Web-Server gesendet wurde, fragt die Web Publishing Engine die FileMaker-Datenbank auf der Basis der Query-Befehle und -Parameter ab, die im Stylesheet und im URL definiert sind, und gibt die Daten dann gemäß den Anweisungen im XSLT-Stylesheet aus.


Web-browser
1. XSLT-CWP-Abfrage wird von Web-browser oder Programm an Web-Server gesendet.
Web-Server
Web-Server-Modul
2. Web-Server leitet XSLT-CWP-Abfrage an CWPE.
3. CWPE fordert XML-Daten von Web Publishing Core auf der Basis von Query-Befehlen und -Parametern an.
FileMaker Server Web Publishing Engine
Web Publishing Core
Custom Web Publishing Engine (CWPE)
4. Web Publishing Core fordert Daten von Datenbank an, die der FileMaker-Datenbank-Server bereitstellt.
FileMaker-Datenbank-Server
Produkte.fp7
Kunden.fp7
Bestellungen.fp7
Katalog.fp7
XSLT-CWP-Abfrage wird von Webbrowser oder Programm an FileMaker-Datenbank-Server gesendet.
Daten werden von Datenbank-Server zurückgeleitet und Web Publishing Engine verwendet XSLT-Stylesheet, um Daten umzuwandeln und zu formatieren.
5. FileMaker-Datenbank-Server sendet angeforderte Daten an Web Publishing Core.
6. Web Publishing Core konvertiert Daten in XML und sendet sie an CWPE.
7. CWPE verwendet XSLT-Stylesheet, um die XML-Daten zu formatieren oder umzuwandeln, und sendet die Ausgabe als HTML-Seiten, XML-Dokument oder Text an den Web-Server.
8. Web-Server sendet Ausgabe an Webbrowser oder Programm.

Allgemeine Schritte bei der Verwendung von Custom Web Publishing mit XSLT

Zusammenfassung der Schritte bei der Verwendung von Custom Web Publishing mit XSLT:

1. Stellen Sie in Admin Console sicher, dass XSLT Publishing aktiviert ist. Weitere Informationen finden Sie in der FileMaker Server Hilfe.
2. Öffnen Sie in FileMaker Pro jede FileMaker-Datenbank, die Sie veröffentlichen wollen, und stellen Sie sicher, dass in der Datenbank das erweiterte Zugriffsrecht fmxslt für Custom Web Publishing mit XSLT aktiviert ist. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 19.

   Hinweis Stellen Sie sicher, dass Sie entsprechende FileMaker-Datenbankberechtigungen verwenden, wenn Sie Stylesheets entwickeln, die an den Endbenutzer gegeben werden. Andernfalls haben Sie in der FileMaker-Datenbank eventuell Zugriff auf Layouts und Funktionen, die dann dem Endbenutzer nicht zur Verfügung stehen und damit zu inkonsistentem Verhalten führen.
3. Erstellen Sie XSLT-Stylesheets, die FileMaker-spezifische XSLT-Erweiterungsfunktionen, Query-Befehle und Query-Parameter enthalten, um die XML-Daten aus einer FileMaker-Datenbank zu formatieren oder umzuwandeln.

   Sie können mithilfe des FileMaker XSLT-Site-Assistenten eines oder mehrere grundlegende XSLT-Stylesheets als Ausgangsposition für Ihre Site erstellen. Weitere Informationen finden Sie im nächsten Abschnitt, „Generieren von FileMaker-XSLT-Stylesheets mithilfe des FileMaker XSLT-Site-Assistenten“.

   Sie können auch Ihr eigenes XSLT-Erstellungs- oder Textbearbeitungswerkzeug verwenden, um die XSLT-Stylesheets wie erforderlich zu ändern oder Ihre Stylesheets ganz neu zu entwickeln. Weitere Informationen finden Sie unter Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.
4. Kopieren oder platzieren Sie die XSLT-Stylesheets im Ordner xslt-template-files, der sich im Ordner Web Publishing im Ordner FileMaker Server auf dem Host befindet, auf dem die Web Publishing Engine installiert ist.

   Sie können die Stylesheets auch in einem optionalen Ordner oder einer Ordnerhierarchie innerhalb des Ordners xslt-template-files platzieren.
5. Platzieren Sie etwaige statische Dateien auf dem Web-Server. Weitere Informationen finden Sie unter „Verwenden von FileMaker XSLT-Stylesheets in einer Website oder einem Programm“ auf Seite 48.
6. Erstellen oder ändern Sie eine Website oder ein Programm, das die XSLT-Stylesheets verwendet.

   Sie können z. B. für die Website eine statische Seite wie index.html verwenden, die Web-Benutzer automatisch an ein XSLT-Stylesheet weiterleitet oder einen Link auf das XSLT-Stylesheet enthält.
7. Richten Sie unbedingt Sicherheitsmechanismen für Ihre Site oder Ihr Programm ein.
8. Testen Sie die Site bzw. das Programm mit den XSLT-Stylesheets unter Verwendung der Konten und Berechtigungen, die für Web-Benutzer definiert sind.
9. Geben Sie die Site bzw. das Programm frei und informieren Sie Benutzer.

Generieren von FileMaker-XSLT-Stylesheets mithilfe des FileMaker XSLT-Site-Assistenten

Der FileMaker XSLT-Site-Assistent ist eine Anwendung, mit der Sie grundlegende XSLT-Stylesheets als Ausgangspunkt für Custom Web Publishing mit XSLT erstellen können. Der XSLT-Site-Assistent eignet sich ideal, um den Aufbau von FileMaker XSLT-Stylesheets kennenzulernen. Sie können dann Ihr eigenes Erstellungswerkzeug für XSLT-Stylesheets oder Textverarbeitungswerkzeuge verwenden, um die Stylesheets je nach Bedarf zu verändern. Sie können den XSLT-Site-Assistenten nicht zur Bearbeitung oder Aktualisierung bestehender Stylesheets verwenden, aber Sie können die anfänglichen Stylesheets für eine gesamte Site generieren oder ein einzelnes Stylesheet erzeugen und damit einer bestehenden Site zusätzliche Funktionalität (z. B. das Löschen von Datensätzen) geben.

Mithilfe des XSLT-Site-Assistenten können Sie XSLT-Stylesheets für alle Arten von Seiten generieren, die nützlich für die Arbeit mit FileMaker-Datenbanken über Custom Web Publishing sind. Abhängig von dem im XSLT-Site-Assistenten gewählten Optionen können Sie eine Site erstellen, die Benutzern das Folgende erlaubt:

- Anzeige einzelner Datensätze
- Anzeige einer Liste mit allen Datensätzen in der Datenbank
- Durchsuchen der Datenbank und Anzeigen des Ergebnisses in einer Liste
- Sortieren von Datensätzen
- Hinzufügen von Datensätzen
- Bearbeiten und Duplizieren von Datensätzen
- Löschen von Datensätzen
- Anzeigen eines Statistikberichts

Ferner können Sie eine optionale Homepage generieren, die mit den anderen generierten XSLT-Stylesheet-Seiten verknüpft ist.

Die Web Publishing Engine verwendet jedes Ihrer Stylesheets, um Daten dynamisch aus einer FileMaker-Datenbank zu beziehen, wenn ein Web-Benutzer eine HTTP-Anforderung und einen URL sendet, der eines Ihrer XSLT-Stylesheets referenziert. Die Web Publishing Engine verwendet ein Stylesheet, um die XML-Daten umzuwandeln und zu formatieren, und generiert die resultierende HTML-Seite, mit der der Web-Benutzer arbeiten kann.

Hinweis Die XSLT-Site-Assistenten-Stylesheets transformieren FileMaker-XML-Daten basierend auf der XML-Grammatik fmresultset in HTML-Seiten, so dass die Stylesheets mit anderen Verwendungen von XML-Daten wie dem FileMaker XML-Export nicht kompatibel sind.

Vor der Verwendung des XSLT-Site-Assistenten

Bevor Sie den XSLT-Site-Assistenten verwenden können, um XSLT-Stylesheets für eine Datenbank zu generieren:

- Setzen Sie in der Datenbank das erweiterte Zugriffsrecht fmxslt. Verwenden Sie bei der Ausführung des XSLT-Site-Assistenten Berechtigungen, die denen entsprechen, die Sie den Web-Benutzern geben. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 19.
- Öffnen Sie die Datenbank auf der Datenbank-Server-Komponente von FileMaker Server und geben Sie sie frei. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

- Stellen Sie sicher, dass die Web-Server-Komponente des FileMaker Server-Einsatzes ausgeführt wird.
- Stellen Sie sicher, dass die Web Publishing Engine-Komponente des FileMaker Server-Einsatzes ausgeführt wird.
- Aktivieren Sie XSLT Publishing in der Web Publishing Engine, um die XSLT-Stylesheets zu verwenden und zu testen. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

Starten des XSLT-Site-Assistenten

Hinweis Um den XSLT-Site-Assistenten zu verwenden, muss Java Runtime Environment 5 oder Java Runtime Environment 6 installiert sein.

So starten Sie den XSLT-Site-Assistenten:

1. Öffnen Sie einen Browser mit der Seite FileMaker Server Web Publishing-Tools.

   Wechseln Sie zu folgendem URL.

   http://*<Server>*:16000/tools

   wobei *<Server>* der Rechner ist, auf dem sich FileMaker Server befindet.

2. Klicken Sie auf PHP-Site-Assistent und XSLT-Site-Assistent, um die Seite FileMaker Server Web Publishing-Tools aufzurufen.
3. Klicken Sie auf XSLT-Site-Assistenten starten.

   FileMaker Server installiert die erforderlichen JAR-Dateien auf Ihrem lokalen Rechner und zeigt einen Statusdialog an, bis der Vorgang abgeschlossen ist.

4. (Optional) Nachdem die Dateien installiert sind, können Sie wählen, ob Sie ein Symbol für den XSLT-Site-Assistenten auf Ihrem Desktop installieren möchten. Klicken Sie auf OK, um das Symbol zu installieren.

Sie können jetzt beginnen, den XSLT-Site-Assistenten zu verwenden.

Verwenden des XSLT-Site-Assistenten

Details und umfassende schrittweise Anweisungen für die Verwendung des XSLT-Site-Assistenten finden Sie in der XSLT-Site-Assistent Hilfe. Weitere Informationen über die Verwendung der vom XSLT-Site-Assistenten generierten XSLT-Stylesheets finden Sie unter „Verwenden von FileMaker XSLT-Stylesheets in einer Website oder einem Programm“ auf Seite 48.

Wichtig Wenn Sie bei Verwendung des XSLT-Site-Assistenten eine Datenbank wählen, die mehrere Tabellen enthält, wählen Sie unbedingt Layouts, die mit dieser Tabelle verbunden sind, sonst liefert die generierte Site unerwartete Ergebnisse. Eine Datenbank könnte z. B. die Tabelle „Produkte“ und die Tabelle „Kunden“ enthalten. Wenn Sie die Layouts für eine Seite zum Suchen, eine Seite zur Datensatzbearbeitung und eine Seite zum Hinzufügen von Datensätzen wählen, müssen diese Layouts unbedingt mit derselben Tabelle verknüpft sein.

Über die vom XSLT-Site-Assistenten generierten Stylesheets

Die XSLT-Stylesheets, die der XSLT-Site-Assistent generiert, umfassen mehrere FileMaker-spezifische Verarbeitungsanweisungen, Elemente und Parameter. Einige Beispiele dazu:

- Die Verarbeitungsanweisung <?xslt-cwp-query params="query string-fragment"?> gibt die zu verwendende XML-Grammatik an und definiert den Namen der im XSLT-Site-Assistenten gewählten Datenbank statisch. Weitere Informationen finden Sie unter „Verwenden von statisch definierten Query-Befehlen und Query-Parametern“ auf Seite 56.
- Das Element <xsl:param name="request-query"/> wird verwendet, um auf Query-Informationen in einer Abfrage oder in HTML-Formulardaten zuzugreifen. Mithilfe dieses Elements kann z. B. in den XSLT-Site-Assistent-Stylesheets auf die aktuelle Abfrage-Query-Information zugegriffen werden, um die aktuelle Position in einer Datensatz-Ergebnismenge zu bestimmen und Verknüpfungen zum vorherigen und nächsten Datensatz zu erstellen. Weitere Informationen finden Sie unter „Verwenden der Query-Information in einer Abfrage“ auf Seite 60.
- Das Element <xsl:param name="authenticated-xml-base-uri"/>, das nicht immer eingeschlossen ist, wird verwendet, um auf den authentifizierten Basis-URI in einer Abfrage zuzugreifen, wenn in der Abfrage mehr XML-Daten benötigt werden. Weitere Informationen finden Sie unter „Verwenden des Parameters „authenticated base URI““ auf Seite 62.

Der XSLT-Site-Assistent generiert auch das Stylesheet utilities.xsl zur Definition von Fehlern und gängigen XSLT-Vorlagen, die von mehreren XSLT-Site-Assistent-Stylesheets aufgerufen werden.

Informationen über andere Abschnitte der XSLT-Site-Assistent-Stylesheets finden Sie in Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Verwenden von FileMaker XSLT-Stylesheets in einer Website oder einem Programm

Unabhängig davon, ob Sie XSLT-Stylesheets mithilfe des XSLT-Site-Assistenten generiert oder von Grund auf neu entwickelt haben, sind die Schritte für die Benutzung in einer Website oder einem Programm mit der Web Publishing Engine gleich.

So verwenden Sie FileMaker XSLT-Stylesheets in einer Website oder einem Programm:

1. Kopieren oder platzieren Sie die XSLT-Stylesheets im Ordner xslt-template-files, der sich im Ordner Web Publishing im Ordner FileMaker Server auf dem Host befindet, auf dem die Web Publishing Engine installiert ist.

   Sie können die Stylesheets auch in einem optionalen Ordner oder einer Ordnerhierarchie innerhalb des Ordners xslt-template-files platzieren.

2. Wenn Ihre XSLT-Stylesheets auf statische Dateien verweisen, z. B. statische Bilder oder HTML-Dateien, platzieren Sie die statischen Dateien mit ihrer originalen Ordnerhierarchie innerhalb des Root-Ordners auf dem Web-Server. Stellen Sie sicher, dass der relative Pfad beibehalten wird.

   Nehmen Sie z. B. an, ein XSLT-Stylesheet verweist auf die Bilddatei logo.jpg und verwendet dafür das HTML-Tag <img src="logo.jpg">. Die Datei logo.jpg muss sich auf dem Web-Server an der folgenden Position befinden:

   <root folder>/fmi/xsl/logo.jpg

3. Falls ein Datenbank-Medienfeld einen Dateiverweis anstelle einer tatsächlichen Datei speichert, muss das entsprechende Medienobjekt im Ordner FileMaker Pro Web gespeichert sein, wenn der Datensatz erstellt oder bearbeitet wird, und dann in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software kopiert oder verschoben werden. Weitere Informationen finden Sie unter „Veröffentlichen des Inhalts von Medienfeldern im Web“ auf Seite 21.

   Hinweis Falls das Medienfeld die tatsächlichen Dateien in der FileMaker-Datenbank speichert, ist mit dem Inhalt des Medienfelds keine Aktion erforderlich, wenn die Datenbank richtig auf dem FileMaker Server freigegeben und zugänglich ist.

4. Um ein XSLT-Stylesheet abzufragen und zu verarbeiten, verwenden Sie die folgende URL-Syntax:

   <scheme>://<host>[:<port>]/fmi/xsl/<folder>/<stylesheet>.xsl[?<query string>]

   Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“ auf Seite 52.

Hinweis Für Websites ist es praktisch, ein XSLT-Stylesheet als Homepage zu verwenden, die nicht verlangt, dass Benutzer für den Zugriff einen Query-String eingeben. Der XSLT-Site-Assistent kann eine Datei home.xsl erzeugen, die keinen Query-String verlangt, da sie die Verarbeitungsanweisung <?xslt-cwp-query?> verwendet. Wenn Sie z. B. Ihre Stylesheets (einschließlich eines home.xsl-Stylesheets) in den Ordner my_templates im Ordner xslt-template-files kopiert haben, können Sie folgenden URL verwenden, um die Stylesheets anzufordern und zu verarbeiten:

http://192.168.123.101/fmi/xsl/my_templates/home.xsl

Wichtig Die Web Publishing Engine erlaubt Web-Benutzern nicht, den Quelltext der XSLT-Stylesheets anzuzeigen, die im Ordner xslt-template-files installiert sind. Wenn Web-Benutzer eine Anforderung für die Verarbeitung eines Stylesheets senden, sendet die Web Publishing Engine nur das Ergebnis der Stylesheet-Umwandlung an den Webbrowser bzw. das Programm.

Fehlerbehebung in XSLT-Stylesheets

Wenn Probleme beim Verwenden der XSLT-Stylesheets auftreten, überprüfen Sie das Folgende:

- Die erweiterten Zugriffsrechte in der Datenbank müssen für Custom Web Publishing mit XSLT eingestellt und einem Benutzerkonto zugewiesen sein. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 19.
- Die Datenbank ist auf der Datenbank-Server-Komponente von FileMaker Server bereitgestellt und geöffnet. Weitere Informationen finden Sie in der FileMaker Server Hilfe.
- Prüfen Sie, ob der verwendete Datenbankkontoname und ggf. Ihr Passwort korrekt sind.
- Die Web-Server-Komponente des FileMaker Server-Einsatzes wird ausgeführt.
- Die Web Publishing Engine-Komponente des FileMaker Server-Einsatzes wird ausgeführt.
- XSLT-Publishing muss in der Web Publishing Engine aktiviert sein.
  - Öffnen Sie einen Browser mit der Seite FileMaker Server-Technologietests:

  http://<server>:16000/test

  wobei <server> der Rechner ist, auf dem sich FileMaker Server befindet.
  - Klicken Sie auf den Link XSLT Custom Web Publishing testen, um eine XSLT-Seite zu öffnen, die auf die Testdatenbank FMServer_Sample zugreift.

Weitere Informationen finden Sie im Handbuch *FileMaker Server Einführung* und in der FileMaker Server Hilfe.

Kapitel 6
Entwickeln von FileMaker XSLT-Stylesheets

Dieses Kapitel enthält Informationen über den Aufbau von FileMaker XSLT-Stylesheets und die Verwendung der FileMaker XSLT-Erweiterungsfunktionen.

Verwenden von XSLT-Stylesheets mit der Web Publishing Engine

Beachten Sie beim Entwickeln und Verwenden von XSLT-Stylesheets für die Abfrage von FileMaker XML-Daten über die Web Publishing Engine die folgenden Punkte:

- Um ein XSLT-Stylesheet mit der Web Publishing Engine zu verwenden, müssen Sie den Namen eines XSLT-Stylesheets in einem URL angeben. Wenn Sie kein Stylesheet angeben oder die Web Publishing Engine das Stylesheet nicht findet oder analysiert „parst“, zeigt die Web Publishing Engine eine Fehlerseite an. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“ auf Seite 52.
- Der Stylesheet-Dateiname und der Name des Ordners, in dem das Stylesheet gespeichert ist, müssen URL-kodiertes UTF-8 sein. Falls Ihr Stylesheet mit älteren Webbrowsern kompatibel sein muss, beschränken Sie die Namen auf ASCII-Zeichen.
- Sie müssen die zu verwendende FileMaker XML-Grammatik angeben, entweder als Query-Parameter im URL oder als statisch definierten Query-Parameter in der Verarbeitungsanweisung <?xslt-cwp-query?>. Wenn Sie keine XML-Grammatik angeben, zeigt die Web Publishing Engine einen Fehler an. Weitere Informationen finden Sie unter „Angeben einer XML-Grammatik für ein FileMaker XSLT-Stylesheet“ auf Seite 55.
- Sie können die Query-Parameter zur Identifikation der angeforderten FileMaker XML-Daten im URL oder als statisch definierten Query-Parameter in der Verarbeitungsanweisung <?xslt-cwp-query?> angeben. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“ auf Seite 52 und „Verwenden von statisch definierten Query-Befehlen und Query-Parametern“ auf Seite 56.
- Optional können Sie die Textkodierung einer XSLT-Abfrage mithilfe des Query-Parameters –encoding angeben. Wenn Sie keine Kodierung angeben, verwendet die Web Publishing Engine ihre Standard-kodierungseinstellung für Abfragen. Weitere Informationen finden Sie unter „Festlegen von Text-kodierung für Abfragen“ auf Seite 57.
- Sie können optional eine Ausgabemethode über das method-Attribut des Elements <xsl:output> angeben. Wenn Sie keine Ausgabemethode angeben, verwendet die Web Publishing Engine HTML als Ausgabe. Sie können optional die Kodierung der Ausgabeseite über das encoding-Attribut des Elements <xsl:output> angeben. Wenn Sie keine Kodierung angeben, verwendet die Web Publishing Engine die Standardkodierungseinstellung für Ausgabeseiten. Weitere Informationen finden Sie unter „Angeben einer Ausgabemethode und -kodierung“ auf Seite 58.
- Sie können optional für E-Mail-Nachrichten, die von der Web Publishing Engine gesendet werden, die Textkodierung über einen Funktionsparameter für die Erweiterungsfunktion fmxslt:send_email() angeben. Weitere Informationen finden Sie unter „Senden von E-Mail-Nachrichten von der Web Publishing Engine“ auf Seite 66.

Den Aufbau einer Abfrage beginnt die Web Publishing Engine mit einem beliebigen Query-Befehl und Query-Parametern, die statisch in der optionalen Verarbeitungsanweisung <?xslt-cwp-query?> gespeichert sind. Der statisch definierte Query-Befehl und seine Parameter werden zur Basisabfrage. Die Verarbeitungsanweisung <?xslt-cwp-query?> ist in einem Stylesheet nicht erforderlich, aber ihre Basisabfrage hat Vorrang vor entsprechenden Query-Befehlen oder -Parametern, die im URL-Query-String angegeben sind. Die Web Publishing Engine fügt dann der Basisabfrage im URL-Query-String einen Query-Befehl oder zusätzliche Parameter hinzu, die *nicht* in der Verarbeitungsanweisung <?xslt-cwp-query?> definiert sind. Die Web Publishing Engine bezieht anhand dieser Abfrage die FileMaker XML-Daten und gibt sie in der angegebenen Ausgabemethode bzw. als HTML an Ihren Webbrowser oder Ihr Programm zurück.

Referenz der FileMaker XSLT-Erweiterungsfunktionen

Diese Version enthält eine FileMaker-Datenbank mit dem Namen „XSLT Reference.fp7“, die kurze Beschreibungen und Beispiele für jede FileMaker XSLT-Erweiterungsfunktion bietet. Die Funktionsreferenzdatenbank finden Sie im folgenden Verzeichnis auf jedem Rechner Ihres FileMaker Server 9-Einsatzes (Master- oder Arbeitscomputer).

Mac OS

/Library/FileMaker Server/Example/XSLT

Windows

<Laufwerk>:\Programme\FileMaker\FileMaker Server\Examples\XSLT

wobei <Laufwerk> das Hauptlaufwerk ist, von dem das System startet.

Über die FileMaker XSLT-Starter-Lösung

Diese Version umfasst eine FileMaker XSLT-Starter-Lösung, die ein Beispiel für die Möglichkeiten von XSLT-Lösungen aufzeigt. Die XSLT-Starter-Lösung finden Sie im folgenden Verzeichnis auf jedem Rechner Ihres FileMaker Server 9-Einsatzes (Master- oder Arbeitscomputer).

Mac OS

/Library/FileMaker Server/Example/XSLT/Starter Solution

Windows

<Laufwerk>:\Programme\FileMaker\FileMaker Server\Examples\XSLT\Starter Solution

wobei <Laufwerk> das Hauptlaufwerk ist, von dem das System startet.

Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets

Die URL-Syntax für die Verwendung von FileMaker XSLT-Stylesheets mit der Web Publishing Engine:

<schema>://<host>[:<port>]/fmi/xsl/[<pfad>/]<stylesheet.xsl>[?<query_string>]

Dabei gilt:

- <scheme> kann das HTTP- oder HTTPS-Protokoll sein.
- <host> ist die IP-Adresse oder der Domänenname des Hosts, auf dem der Web-Server installiert ist.
- <port> ist optional und gibt den Port an, den der Web-Server überwacht. Ohne Portangabe wird der Standard-Port des Protokolls (Port 80 für HTTP oder Port 443 für HTTPS) verwendet.

- <path> ist optional und gibt den/die Ordner innerhalb des Ordners xslt-template-files an, in dem sich das XSLT-Stylesheet befindet.
- <stylesheet.xsl> ist der Dateiname des XSLT-Stylesheets.
- <query_string> kann eine Kombination aus einem Query-Befehl und einem oder mehreren Query-Parametern für Custom Web Publishing mit XSLT sein. Weitere Informationen finden Sie unter „Verwenden von Query-Strings in FileMaker XSLT-Stylesheets“ auf Seite 54, und in Anhang A, „Gültige Namen in Query-Strings“. Wenn das angegebene Stylesheet eine <?xslt-cwp-query?>-Verarbeitungsanweisung enthält, haben der statisch definierte Query-Befehl und die Parameter Vorrang vor anderen Query-Befehlen oder Parametern im URL-Query-String. Weitere Informationen finden Sie unter „Verwenden von statisch definierten Query-Befehlen und Query-Parametern“ auf Seite 56.

Hinweis Die URL-Syntax einschließlich der Namen von Query-Befehl und -Parametern unterscheidet Groß- und Kleinschreibung, mit Ausnahme von Teilen des Query-Strings. Der URL ist vorwiegend in Kleinbuchstaben mit Ausnahme der Grammatiknamen in Großbuchstaben: FMPXMLRESULT und FMPXMLLAYOUT. Informationen über die Regeln für Groß-/Kleinschreibung im Query-String finden Sie unter „Richtlinien für die Verwendung von Query-Befehlen und -Parametern“ auf Seite 88.

Beispiel eines URL für die Verwendung eines FileMaker XSLT-Stylesheets mit der Web Publishing Engine:

http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xsl?–grammar=fmresultset&–db=mydatabase&–lay=mylayout&–findall

Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen

In einem generierten XML-Dokument für eine XSLT-Lösung unterscheidet sich die Syntax für den Verweis auf ein Medienobjekt für Medienfelder, die das tatsächliche Objekt in der Datenbank speichern, und für Medienfelder, die einen Verweis auf das Objekt speichern.

- Wenn ein Medienfeld das tatsächliche Objekt in der Datenbank speichert, dann verwendet das <data>-Element des Medienfelds die folgende URL-Syntax für den Verweis auf das Objekt:

  <data>/fmi/xsl/cnt/data.<extension>?<query string></data>.

  Dabei gilt: <extension> ist die Dateinamenerweiterung zur Identifizierung des Objekttyps wie beispielsweise .jpg oder .mov. Informationen über <query string> finden Sie im vorherigen Abschnitt, “Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets.”

  Beispiel:

  <data>/fmi/xsl/cnt/data.jpg?–db=products&–lay=sales&–field=product_image(1)&–recid=2</data>

  Hinweis In der generierten XML für ein Medienfeld ist der Wert für den Query-Parameter –field ein vollständig qualifizierter Feldname. Die Zahl in Klammern gibt die Wiederholungsnummer für das Medienfeld an und wird für Wiederhol- und für Einzelfelder generiert. Weitere Informationen finden Sie unter „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 89.

  Verwenden Sie zum Abrufen der Mediendaten aus der Datenbank die folgende Syntax:

  <schema>://<host>[:<port>]/fmi/xsl/cnt/data.<erweiterung>?<query_string>

  Informationen über <schema>, <host> oder <port> finden Sie im vorherigen Abschnitt, “Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets.”

Beispiel:

http://www.company.com/fmi/xsl/cnt/data.jpg?–db=products&–lay=sales&–field=product_image(1)&–recid=2

- Wenn ein Medienfeld einen Dateiverweis anstelle eines tatsächlichen Objekts speichert, enthält das <data>-Element des Medienfelds einen relativen Pfad, der auf das Objekt verweist. Wenn sich z. B. logo.jpg im Web-Ordner innerhalb des FileMaker Pro-Ordners befindet, lautet das <data>-Element des Medienfelds:

  <data>/images/logo.jpg</data>

  Hinweis Das Medienobjekt, auf das verwiesen wird, muss im Ordner FileMaker Pro Web gespeichert sein, wenn der Datensatz erstellt oder bearbeitet wird, und dann in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software kopiert oder verschoben werden. Weitere Informationen finden Sie unter „Veröffentlichen des Inhalts von Medienfeldern im Web“ auf Seite 21.

- Für ein leeres Medienfeld ist auch das <data>-Element leer.

Verwenden von Query-Strings in FileMaker XSLT-Stylesheets

Beim Verwenden eines Query-Strings in einem URL oder der Verarbeitungsanweisung <?xslt-cwp-query?> in einem FileMaker XSLT-Stylesheet können Sie beliebige Query-Befehle und -Parameter einschließen, die zur Abfrage von XML-Daten aus einer FileMaker-Datenbank definiert sind. Weitere Informationen finden Sie unter „Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings“ auf Seite 38.

Sie können auch den folgenden Query-Befehl und Parameter benutzen, die nur zur Verwendung mit FileMaker XSLT-Stylesheets definiert sind.

Verwenden Sie diesen XSLT Query-Befehl oder Parameternamen	Für	Kommentar
–grammar	Angeben der XML-Grammatik für XSLT-CWP-Abfragen oder für XSLT-Stylesheets. Weitere Informationen finden Sie im nächsten Abschnitt, “Angeben einer XML-Grammatik für ein FileMaker XSLT-Stylesheet.”.	Dieser Query-Parameter ist in allen XSLT-Abfragen erforderlich.
–encoding	Angeben der Textkodierung für eine Abfrage. Weitere Informationen finden Sie unter „Festlegen von Textkodierung für Abfragen“ auf Seite 57.	Dieser Query-Parameter ist in allen XSLT-Abfragen optional.
–process	Verarbeitung eines Stylesheets ohne Datenabfrage. Weitere Informationen finden Sie unter „Verarbeitung von XSLT-Abfragen, die nicht FileMaker Server abfragen“ auf Seite 58.	Dieser Query-Befehl erfordert den Query-Parameter –grammar.
–token	Weitergeben von Werten zwischen Seiten, ohne Sitzungen oder Cookies zu verwenden. Weitere Informationen finden Sie unter „Verwendung von Tokens zur Übergabe von Informationen zwischen Stylesheets“ auf Seite 59.	Dieser Query-Parameter ist in allen XSLT-Abfragen optional.

Angeben einer XML-Grammatik für ein FileMaker XSLT-Stylesheet

Die empfohlene XML-Grammatik für Custom Web Publishing mit XSLT ist die Grammatik fmresultset, die zur bequemen Benutzung mit XSLT entwickelt wurde. Weitere Informationen finden Sie unter „Verwenden der fmresultset-Grammatik“ auf Seite 32. Sie können auch die älteren Grammatiken FMPXMLRESULT oder FMPXMLLAYOUT verwenden. Für den Zugriff auf Wertelisten und Feldanzeigedaten in Layouts müssen Sie die FMPXMLLAYOUT-Grammatik verwenden. Weitere Informationen finden Sie unter „Verwenden anderer FileMaker XML-Grammatiken“ auf Seite 35. Die Grammatik FMPDSORESULT kann nicht mit Custom Web Publishing mit XSLT verwendet werden.

Um die Grammatik für ein FileMaker XSLT-Stylesheet anzugeben, verwenden Sie den Query-Parameter –grammar in einem URL oder den statisch definierten Query-Parameter in der Verarbeitungsanweisung <?xslt-cwp-query?>.

In einem URL:

```
http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xsl?–grammar=fmresultset&–db=mydatabase
&–lay=mylayout&–findall
```

In einer Verarbeitungsanweisung:

```
<?xslt-cwp-query params="–grammar=fmresultset&–db=mydatabase&–lay=mylayout&–findall"?>
```

Wichtig Wenn Sie keine XML-Grammatik für ein FileMaker XSLT-Stylesheet angeben, wird der Fehler „QUERY -ER0001“ angezeigt. Weitere Informationen finden Sie unter Anhang B, „Fehlercodes für Custom Web Publishing“.

Erläuterung von Namespaces und Präfixen für FileMaker XSLT-Stylesheets

Mithilfe eindeutiger XSLT-Namespaces können Sie XSLT-Tags der richtigen Anwendung zuordnen. Deklarieren Sie im Element <xsl:stylesheet> am Beginn aller FileMaker XSLT-Stylesheets die Namespaces für die FileMaker XSLT-Erweiterungsfunktionen und die jeweiligen Grammatiken, die Sie in dem Stylesheet verwenden.

Verwendung	Namespace deklarieren	Präfix verwenden
fmresultset-XML-Grammatik	xmlns:fmrs="http://www.filemaker.com/xml/fmresultset"	fmrs
FMPXMLRESULT-Grammatik	xmlns:fmp="http://www.filemaker.com/fmpxmlresult"	fmp
FMPXMLLAYOUT-Grammatik	xmlns:fml="http://www.filemaker.com/fmpxmllayout"	fml
Für die Query-XML-Grammatik	xmlns:fmq="http://www.filemaker.com/xml/query"	fmq
Für die FileMaker XSLT-Erweiterungsfunktionen	xmlns:fmxslt="xalan://com.fmi.xslt.ExtensionFunctions"	fmxslt

Sie müssen auch den folgenden erforderlichen Namespace in jedem FileMaker XSLT-Stylesheet deklarieren:

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
```

Beispiel für Deklarationen von Namespaces:

```
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:fmrs="http://www.filemaker.com/xml/fmresultset"
    xmlns:fml="http://www.filemaker.com/fmpxmllayout"
    xmlns:fmq="http://www.filemaker.com/xml/query"
    xmlns:fmxslt="xalan://com.fmi.xslt.ExtensionFunctions"
    exclude-result-prefixes="xsl fmrs fmq fml fmxslt">
```

Verwenden von statisch definierten Query-Befehlen und Query-Parametern

Sie können mit Ihrem FileMaker XSLT-Stylesheet die unbefugte Verwendung von Query-Befehlen und Query-Parametern verhindern, indem Sie die zu verwendenden Query-Befehle und -Parameter beim Abfragen von XML-Daten statisch definieren. Wenn Query-Befehle und -Parameter statisch in einem Stylesheet definiert sind (was nicht erforderlich ist), haben sie Vorrang vor etwaigen übereinstimmenden Query-Befehlen oder -Parametern, die ein URL-Query-String angibt.

Die Stylesheets, die von dem XSLT-Site-Assistenten generiert werden, verwenden statisch definierte Query-Befehle und -Parameter. FileMaker empfiehlt die Verwendung statisch definierter Query-Befehle und -Parameter als beste Technik für eine größere Sicherheit Ihrer Lösung.

Um Query-Befehle und -Parameter statisch zu definieren, verwenden Sie die folgende Verarbeitungsanweisung am Beginn Ihres FileMaker XSLT-Stylesheets:

```
<?xslt-cwp-query params="query string-fragment"?>
```

Dabei gilt:

query string-fragment ist ein String mit den Name-Wert-Paaren in folgendem Format:

```
name=wert&name2=wert2....
```

Dabei gilt:

name ist ein String, der den Namen eines Query-Befehls, Query-Parameters oder Datenbankfelds angibt.

wert ist ein Stringwert beliebiger Länge. Verwenden Sie für Query-Parameter und Feldnamen den jeweiligen Wert, den Sie definieren wollen, z. B. –db=Produkte. Geben Sie für Query-Befehle kein =-Zeichen oder einen Wert nach dem Befehlsnamen wie –findall an. Weitere Informationen finden Sie unter Anhang A, „Gültige Namen in Query-Strings“.

Die im Fragment verwendeten Strings müssen URL-kodiert sein. Weitere Informationen finden Sie unter „Erläuterung von URL-Textkodierung“ auf Seite 29. Sie müssen dieselbe Zeichenkodierung verwenden, die das encoding-Attribut im Tag <xsl:output> angibt. Wenn keine Kodierung angegeben wird, verwendet die Web Publishing Engine ihre konfigurierte Standardkodierung.

Das Trennzeichen zwischen den beiden Name-Wert-Paaren muss ein kaufmännisches Und-Zeichen (&) sein.

Nehmen Sie z. B. an, Sie haben die folgende Verarbeitungsanweisung in einem Stylesheet mit dem Namen my_stylesheet.xsl verwendet:

```
<?xslt-cwp-query params="–db=products&–lay=sales&–grammar=fmresultset&productname=the%20item&–find"?>
```

Diese Beispiel-Verarbeitungsanweisung würde alle Abfragen an my_stylesheet.xsl zwingen, die fmresultset-Grammatik mit der Datenbank products und dem Layout sales zu verwenden und eine -find-Abfrage mit dem Wert the%20item im Feld „Produktname“.

Nun führt ein Client die folgende Abfrage mit my_stylesheet.xsl durch:

```
http://server.company.com/fmi/xsl/my_stylesheet.xsl?–lay=revenue&city=London&–edit
```

Die Web Publishing Engine verarbeitet in diesem Fall die folgende XML-Abfrage:

```
http://server.company.com/fmi/xml/fmresultset.xml?–db=products&–lay=sales&productname=the%20item&city=
London&–find
```

Der statisch definierte Query-Befehl und seine Parameter setzen den Query-Parameter –lay=revenue und den Query-Befehl –edit außer Kraft, die der Client angegeben hat. Da das Feld „city“ in der Verarbeitungsanweisung nicht statisch definiert war, übernimmt die Web Publishing Engine dafür in der XML-Abfrage den Wert London, den der Client eingegeben hat.

Festlegen von Textkodierung für Abfragen

Die Web Publishing Engine führt die folgenden Schritte in der gezeigten Reihenfolge aus, bis sie die Kodierung einer XSLT-Abfrage ermittelt:

1. Prüfen, ob das charset-Attribut im Abfrage-Header „Content-Type“ gesetzt ist.
2. Prüfen, ob Sie eine Kodierung mit dem Query-Parameter –encoding angegeben haben. Sie können diesen Parameter in einem URL oder als statisch definierten Query-Parameter in der Verarbeitungsanweisung <?xslt-cwp-query?> angeben. Der Wert des –encoding-Parameters gibt die Kodierung an, die für die übrigen Parameter in der Abfrage verwendet wird. Die gültigen Werte für diesen Parameter werden in der folgenden Tabelle aufgelistet. Beispiel:

   http://192.168.123.101/fmi/xsl/template/my_stylesheet.xsl?–db=products–lay=sales&–grammar=fmresultset&–encoding=Shift_JIS&–findall

3. Verwenden der aktuellen Einstellung für die Standard-Textkodierungsoption der Abfrage- und Ausgabeseiten für die Web Publishing Engine. Direkt nach der Installation der Web Publishing Engine ist die Textkodierung für Abfragen standardmäßig auf UTF-8 eingestellt. Sie können die Textkodierungseinstellungen der Web Publishing Engine über Admin Console ändern. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

Nachdem die Web Publishing Engine die Kodierung ermittelt, wird diese Kodierung verwendet und es werden keine weiteren Schritte zur Ermittlung der Kodierung unternommen. Wenn z. B. das charset-Attribut im Abfrage-Header „Content-Type“ gesetzt ist, verwendet die Web Publishing Engine nicht den Wert des Query-Parameters –encoding.

Die Textkodierung, die über eine der obigen Methoden angegeben wird, muss eine der folgenden Kodierungen verwenden:

Kodierung	Beschreibung
US-ASCII	Der ASCII-Basiszeichensatz wird typischerweise für englische E-Mails mit Standardtext verwendet.
ISO-8859-1	Der Latin 1-Zeichensatz wird typischerweise für Web-Seiten und E-Mail-Nachrichten mit lateinischen Buchstaben aus dem oberen ASCII-Zeichenbereich verwendet.
ISO-8859-15	Der Latin 9-Zeichensatz, der beinahe mit dem Latin 1-Zeichensatz übereinstimmt, umfasst zusätzlich das Euro-Symbol €.
ISO-2022-JP	Die ISO Japanese-Kodierung wird typischerweise für japanische E-Mail-Nachrichten verwendet.
Shift_JIS	Diese Japanisch-Kodierung wird typischerweise für japanische Web-Seiten verwendet.
UTF-8	Die Acht-Bit-Kodierung von Unicode. Die Verwendung von UTF-8 für E-Mail-Nachrichten und Web-Seiten gewinnt an Beliebtheit, da gängige Browser und E-Mail-Clients diese Kodierung jetzt unterstützen. Da UTF-8 den kompletten Bereich an Unicode-Zeichen unterstützt, sind damit Seiten in jeder Sprache möglich.

Notizen

- Direkt nach der Installation der Web Publishing Engine ist die Textkodierung für Ausgabeseiten standardmäßig auf UTF-8 eingestellt. Weitere Informationen finden Sie im nächsten Abschnitt, “Angeben einer Ausgabemethode und -kodierung.” Für E-Mail-Nachrichten verwendet die Web Publishing Engine anfangs für die Textkodierung ISO-8859-1 als Standardeinstellung. Sie können diese Einstellung über Admin Console ändern.
- Sie können auch die E-Mail-Nachrichtenkodierung mithilfe der Erweiterungsfunktion fmxslt:send_email(String smtpFields, String body, String encoding) festlegen. Weitere Informationen finden Sie unter „Senden von E-Mail-Nachrichten von der Web Publishing Engine“ auf Seite 66.

Angeben einer Ausgabemethode und -kodierung

Sie können eine Ausgabemethode und -kodierung der Ausgabeseiten mithilfe des method- und encoding-Attributs des Elements <xsl:output> angeben. Beide Attribute sind optional.

Das method-Attribut legt den Typ der Ausgabe fest, d. h. „html“, „text“ oder „xml“. Andere Methodentypen werden nicht unterstützt. Wenn Sie keine Methode angeben, verwendet die Web Publishing Engine „html“.

Das encoding-Attribut gibt die Kodierung von Ausgabeseiten an. Sie können eine beliebige Kodierung angeben, die in der Tabelle im vorherigen Abschnitt aufgeführt ist. Wenn Sie keine Kodierung angeben, verwendet die Web Publishing Engine ihre Standardkodierungseinstellung für Ausgabeseiten.

Beispiel:

```
<xsl:output method="html" encoding="ISO-8859-1"/>
```

Wenn Sie das Element <xsl:output> nicht in einem Stylesheet verwenden, gibt die Web Publishing Engine HTML-Seiten in der aktuellen Standardeinstellung für Textkodierung von Ausgabeseiten aus.

Erläuterung der Kodierung von XSLT-Stylesheets

Neben der Kodierung für Abfragen und Ausgabeseiten muss die Kodierung Ihrer XSLT-Stylesheets auch im encoding-Attribut der XML-Deklaration am Beginn des Stylesheets angegeben werden. Sie können eine beliebige Kodierung angeben, aufgeführt in der Tabelle auf Seite 57.

Beispielsweise gibt die folgende Deklaration UTF-8 als Kodierung des Stylesheets an:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Wenn Sie keine Stylesheet-Kodierung angeben, nimmt die Web Publishing Engine UTF-8 als Kodierung an.

Verarbeitung von XSLT-Abfragen, die nicht FileMaker Server abfragen

Sie können mithilfe des Query-Befehls –process XSLT-Abfragen verarbeiten, wenn diese keine Daten aus der Datenbank benötigen oder wenn Ihr Stylesheet keine datenbankspezifischen Informationen wie Datensätze, Feldnamen oder Layoutnamen erfordert. Indem Sie in diesen Fällen den Befehl –process verwenden, können Sie die Arbeitslast von FileMaker Server verringern.

Mithilfe des Befehls –process können Sie z. B.:

- ein Stylesheet laden, das eine statische Seite generiert, wenn keine Datenbankinformation benötigt wird.
- ein Stylesheet laden, das einen neuen Datensatz erstellt, wenn das Stylesheet keine Datenbank- oder Layoutinformation wie eine Werteliste benötigt.

- eine Erweiterungsfunktion wie fmxslt:send_email() verwenden, die keine Daten aus der Datenbank benötigt.
- auf Informationen zugreifen, die in einer Sitzung gespeichert sind, wenn keine Datenbankinformation benötigt wird.

Der Befehl –process gibt ein XML-Dokument zurück, das Produktinformationen über die Web Publishing Engine enthält.

Der einzige erforderliche Parameter für den Befehl –process ist –grammar und Sie müssen die fmresultset- oder die FMPXMLRESULT-Grammatik verwenden.

Beispiel:

http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xsl?–grammar=fmresultset&–process

Verwendung von Tokens zur Übergabe von Informationen zwischen Stylesheets

Sie können den Query-Parameter –token in einem URL oder als statisch definierten Query-Befehl verwenden, um beliebige benutzerdefinierte Informationen ohne Sitzungen oder Cookies zwischen Stylesheets zu übergeben. Der Query-Parameter –token ist in allen Query-Befehlen optional.

Der benutzerdefinierte Parameterwert kann ein beliebiger URL-kodierter String sein. Beispiel:

http://192.168.123.101/fmi/xsl/template/my_stylesheet.xsl?–db=products&–lay=sales&–grammar=fmresultset&–token.D100=Pending&–findall

Weitere Informationen finden Sie unter „Query-Parameter –token.[string] (Werte zwischen XSLT-Stylesheets übergeben)“ auf Seite 105.

Wichtig Verwenden Sie den Query-Parameter –token nicht, um private Daten zu übergeben.

Um den Wert des Query-Parameters –token abzurufen, verwenden Sie die Anweisung <xsl:param name="request-query"/>. Weitere Informationen finden Sie unter „Verwenden der Query-Information in einer Abfrage“ auf Seite 60.

Verwenden der FileMaker XSLT-Erweiterungsfunktionen und -parameter

Die FileMaker XSLT-Erweiterungsfunktionen befinden sich laut Definition im fmxslt-Namespace. Stellen Sie sicher, dass Sie eine Deklaration des fmxslt-Namespace im Element <xsl:stylesheet> am Beginn Ihres XSLT-Stylesheets angeben. Weitere Informationen finden Sie unter „Erläuterung von Namespaces und Präfixen für FileMaker XSLT-Stylesheets“ auf Seite 55.

Die FileMaker XSLT-Erweiterungsfunktionen wurden so entwickelt, dass Sie sie in einem XSLT-Stylesheet verwenden können, indem Sie sie als Funktionsaufruf in einer XPath-Anweisung angeben. XPath-Anweisungen werden als Werte des select-Attributs und des test-Attributs in zahlreichen XSLT-Elementen verwendet.

Nehmen Sie z. B. an, Sie wollen im User-Agent-Header feststellen, welcher Browser benutzt wird. Dafür sollten Sie eine Variable verwenden, die den Wert des User-Agent-Headers enthält:

```
<xsl:variable name="user-agent" select="fmxslt:get_header('User-Agent')"/>
```

Für die Erweiterungsfunktionen, die einen Wert zurückgeben, weist der Wert den angegebenen XSLT-Typ auf. Viele Funktionen geben Strings zurück, aber einige Funktionen geben eine Knotengruppe (node-set) zurück, die durchlaufen werden kann.

Hinweis Dieser Abschnitt beschreibt die FileMaker XSLT-Erweiterungsfunktionen und -parameter und bietet einige Beispiele. Zusätzliche Beispiele für jede Funktion finden Sie in der Referenz der FileMaker XSLT-Erweiterungsfunktionen. Weitere Informationen finden Sie unter „Referenz der FileMaker XSLT-Erweiterungsfunktionen“ auf Seite 52.

Erläuterung der FileMaker-spezifischen XSLT-Parameter, die von der Web Publishing Engine gesetzt werden

Beim Verarbeiten einer Abfrage setzt die Web Publishing Engine dynamisch die Werte der folgenden FileMaker-spezifischen XSLT-Parameter. Sie können die Werte dieser Parameter in Ihrem Stylesheet mithilfe des Elements <xsl:param> verwenden.

FileMaker-spezifischer XSLT-Parameter	Weitere Informationen
<xsl:param name="request-query"/>	“Verwenden der Query-Information in einer Abfrage” im nächsten Abschnitt.
<xsl:param name="client-ip"/> <xsl:param name="client-user-name"/> <xsl:param name="client-password"/>	„Abrufen von Clientinformationen“ auf Seite 61.
<xsl:param name="xml-base-uri"/>	„Verwenden des Basis-URI-Parameters der Web Publishing Engine“ auf Seite 61.
<xsl:param name="authenticated-xml-base-uri">	„Verwenden des Parameters „authenticated base URI““ auf Seite 62.

Verwenden der Query-Information in einer Abfrage

Sie können mithilfe eines FileMaker XSLT-Parameters auf Informationen in einer Abfrage in einem URL oder in HTML-Formulardaten zugreifen. Beispielsweise können Sie auf die aktuelle Abfrage-Query-Information zugreifen, um die aktuelle Position in einer Datensatz-Ergebnismenge zu bestimmen und Verknüpfungen zum vorherigen und nächsten Datensatz zu erstellen.

Die folgenden FileMaker XSLT-Parameter bieten Zugriff auf alle Query-Befehle und Query-Parameter, die für Abfragen von FileMaker XML-Daten über die Web Publishing Engine verwendet werden:

<xsl:param name="request-query"/>

Mit Ausnahme von Feldnamen gibt die Web Publishing Engine alle Namen von Query-Befehlen und Query-Parametern in Kleinbuchstaben zurück. Die Groß-/Kleinschreibung von Feldnamen wird beibehalten.

Ein XML-Dokumentfragment wird in den Abfrage-Query-Parameter in der folgenden Grammatik geladen:

```
<!DOCTYPE query [
    <!ELEMENT query (parameter)*>
        <!ATTLIST query action CDATA #REQUIRED>
        <!ELEMENT parameter (#PCDATA)>
        <!ATTLIST parameter name CDATA #REQUIRED>
        ]
```

Hinweis Die Query-Information befindet sich laut Definition im Namespace fmq="http://www.filemaker.com/xml/query". Stellen Sie sicher, dass Sie eine Deklaration des fmq-Namespace im Element <xsl:stylesheet> am Beginn Ihres XSLT-Stylesheets angeben. Weitere Informationen finden Sie unter „Erläuterung von Namespaces und Präfixen für FileMaker XSLT-Stylesheets“ auf Seite 55.

Nehmen Sie z. B. an, Sie wollen auf die Query-Befehle und Query-Parameter in dieser Abfrage zugreifen:

http://192.168.123.101/fmi/xsl/my_stylesheet.xsl?–db=products&–lay=sales&–grammar=fmresultset&–token.1=abc123&–findall

Wenn Sie die Anweisung <xsl:param name="request-query" /> vor dem Vorlagenabschnitt (template) einfügen, speichert die Web Publishing Engine dieses XML-Dokumentfragment in diesem Parameter:

```
<query action="my_stylesheet.xsl" xmlns="http://www.filemaker.com/xml/query">
    <parameter name="–db">products</parameter>
    <parameter name="–lay">sales</parameter>
    <parameter name="–grammar">fmresultset</parameter>
    <parameter name="–token.1">abc123</parameter>
    <parameter name="–findall"></parameter>
</query>
```

Sie können dann den Parameter request-query verwenden, um auf den Wert eines Tokens zuzugreifen, der über einen XPath-Ausdruck in einem URL übergeben wurde. Beispiel:

```
$request-query/fmq:query/fmq:parameter[@name = '–token.1']
```

Abrufen von Clientinformationen

Sie können mithilfe der folgenden FileMaker XSLT-Parameter Informationen über die IP-Adresse, den Benutzernamen und das Passwort eines Web-Clients von der Web Publishing Engine abrufen:

```
<xsl:param name="client-ip"/>
<xsl:param name="client-user-name"/>
<xsl:param name="client-password">
```

Nehmen Sie diese Parameteranweisungen vor dem ersten <xsl:template>-Element in Ihr XSLT-Stylesheet auf.

Diese Parameter identifizieren den Web-Benutzer, wenn ein Stylesheet programmtechnisch weitere passwortgeschützte XML-Dokumente lädt. Weitere Informationen finden Sie unter „Laden zusätzlicher Dokumente“ auf Seite 62. Der Web-Benutzer muss den Benutzernamen und das Passwort anfangs im Dialogfeld „HTTP-Standardauthentifizierung“ angeben. Weitere Informationen finden Sie unter „Zugriff auf eine geschützte Datenbank“ auf Seite 19.

Weitere Informationen und Beispiele für die Verwendung dieser drei FileMaker XSLT-Parameter finden Sie in der Referenz der FileMaker XSLT-Erweiterungsfunktionen.

Verwenden des Basis-URI-Parameters der Web Publishing Engine

Die Web Publishing Engine definiert den Basis-Uniform Resource Identifier (URI)-Parameter als Host und Port, an dem die Web Publishing Engine installiert ist. Dieser Basis-URI ermöglicht, dass Abfragen von XML-Daten aus FileMaker-Datenbanken in Relation zum Web Publishing Engine-Host aufgelöst werden.

Um auf den Basis-URI der Web Publishing Engine zuzugreifen, nehmen Sie diese Anweisungen vor dem ersten <xsl:template>-Element in Ihr XSLT-Stylesheet auf:

```
<xsl:param name="xml-base-uri"/>
```

Sie können den Basis-URI dann für das aktuelle Stylesheet über die Variable $xml-base-uri immer dann verwenden, wenn Sie eine zusätzliche Abfrage auf FileMaker XML-Daten ausführen müssen. Sie können z. B. den Basis-URI in der folgenden Abfrage von zusätzlichen XML-Daten verwenden:

```
<xsl:variable name="layout_information" select="document(concat($xml-base-uri,'/fmi/xml/FMPXMLLAYOUT.xml?
–db=products&–lay=sales&–view'))" />
```

Verwenden des Parameters „authenticated base URI"

Der Parameter authenticated-xml-base-uri kombiniert die Funktionalität der Parameter client-user-name und client-password mit dem Parameter xml-base-uri:

```
<xsl:param name="authenticated-xml-base-uri"/>
```

Verwenden Sie diesen Parameter, um ein zusätzliches, passwortgeschütztes XML-Dokument zu laden, das den Benutzernamen und das Passwort verlangt, das anfangs in der aktuell verarbeiteten Abfrage eingegeben wurde. Ein Beispiel finden Sie im nächsten Abschnitt, "Laden zusätzlicher Dokumente."

Nehmen Sie diese Parameteranweisung vor dem ersten <xsl:template>-Element in Ihr XSLT-Stylesheet auf.

Wenn die Werte der Parameter client-user-name und client-password nicht leer sind, lautet der Wert des Parameters authenticated-xml-base-uri:

```
http://username:password@hostname:port
```

Wenn die Werte der Parameter client-user-name und client-password leer sind, ist der Wert des Parameters authenticated-xml-base-uri identisch mit dem Wert des Parameters xml-base-uri.

Laden zusätzlicher Dokumente

Um bei der Verarbeitung eines XSLT-Stylesheets ein zusätzliches XML-Dokument zu laden, verwenden Sie die XSLT-Standardfunktion document() mit einem URI für das XML-Dokument. Die Funktion document() gibt die angeforderte XML als eine Knotengruppe (node-set) zurück, die in einem <xsl:variable>-Element gespeichert werden kann.

Um ein XML-Dokument zu laden, das Daten aus einer FileMaker-Datenbank enthält, verwenden Sie die Funktion document() mit FileMaker-Query-Befehl und -Parametern. Beispiel:

```
<xsl:variable name="other-data" select="document(concat($xml-base-uri,'/fmi/xml/FMPXMLLAYOUT.xml?
–db=products&–lay=sales&–view'))" />
```

Um ein zusätzliches, passwortgeschütztes XML-Dokument zu laden, das den Benutzernamen und das Passwort verlangt, das ursprünglich in der aktuell verarbeiteten Abfrage eingegeben wurde, verwenden Sie den Parameter authenticated-xml-base-uri. Dieser Parameter gibt denselben Benutzernamen und dasselbe Passwort als Teil des URI an, der an die Funktion document() übergeben wird.

Beispiel:

```
<xsl:variable name="other-data" select="document(concat($authenticated-xml-base-uri,
'/fmi/xml/FMPXMLLAYOUT.xml?–db=products&–lay=sales&–view'))"/>
```

Um ein passwortgeschütztes XML-Dokument zu öffnen, das einen anderen Benutzernamen und ein anderes Passwort verlangt, als in der übergeordneten Abfrage angegeben wurde, geben Sie anhand der folgenden Syntax den Benutzernamen und das Passwort als Teil des URI an, der an die Funktion document() übergeben wurde:

```
http://username:password@hostname/path?querystring
```

Um ein XML-Dokument zu laden, das nicht auf einer FileMaker-Datenbank basiert, verwenden Sie die Funktion document() ohne FileMaker-Query-Befehle oder Query-Parameter. Beispiel:

```
<xsl:variable name="other-data" select="document('http://server.company.com/data.xml')" />
```

Wenn Sie die Funktion document() mit einem relativen URL verwenden, versucht die Web Publishing Engine, das XML-Dokument von der Position im lokalen Dateisystem zu laden, die dem relativen Speicherort des Stylesheets entspricht. Nehmen Sie z. B. an, dass ein Stylesheet, das sich im Ordner mystylesheets innerhalb des Ordners xslt-template-files befindet, die folgende document()-Funktion mit einem relativen URL enthält:

```
<xsl:variable name="mydoc" select="document('mystylesheets/mydoc.xml')" />
```

Die Web Publishing Engine versucht, mydoc.xml aus dem Ordner mystylesheets im Ordner xslt-template-files im lokalen Dateisystem zu laden.

Hinweis Wenn Sie den Basis-URI der Web Publishing Engine verwenden, um ein Dokument zu laden, unterstützt die Web Publishing Engine nur HTTP. Wenn Sie ein Dokument von einem externen Server laden, unterstützt die Web Publishing Engine sowohl HTTP als auch HTTPS.

Verwenden der Layoutinformationen für eine Datenbank in einem Stylesheet

Sie können die Layoutinformation einer FileMaker-Datenbank in ein Stylesheet übernehmen, indem Sie die Information mit der FMPXMLLAYOUT-Grammatik abfragen und dann über die XSLT-Funktion document() in eine Variable laden:

```
<xsl:variable name="layout" select="document(concat($xml-base-uri,'/fmi/xml/FMPXMLLAYOUT.xml?–view'))" />
```

Nehmen Sie z. B. an, Sie wollen ein Menü für ein Feld mit dem Namen Color erstellen, das mit den Werten aus einer Werteliste mit dem Namen shirts gefüllt ist, die wiederum in einem Layout in einer FileMaker-Datenbank definiert sind. So können Sie die Layoutinformation mithilfe der Funktion document() in eine XSLT-Variable laden:

```
<xsl:variable name="layout" select="document(concat($xml-base-uri,'/fmi/xml/FMPXMLLAYOUT.xml?–db=products
&–lay=sales&–view'))" />
<select size="1">
    <xsl:attribute name="name">Color</xsl:attribute>
    <option value="">Select One...</option>
    <xsl:for-each select="$layout/fml:FMPXMLLAYOUT/fml:VALUELISTS/fml:VALUELIST[@NAME = 'shirts']/
    fml:VALUE">
        <option:>
            <xsl:attribute name="value"><xsl:value-of select="."/></xsl:attribute>
            <xsl:value-of select="."/>
        </option>
    </xsl:for-each>
</select>
```

Verwenden von Inhaltspufferung

Wenn Inhaltspufferung deaktiviert ist, streamt die Web Publishing Engine das Ergebnis einer XSLT-Transformation direkt zurück an den Client. Inhaltspufferung ist immer deaktiviert, sofern Sie sie nicht explizit aktivieren. Wenn Sie Inhaltspufferung aktivieren, speichert die Web Publishing Engine den umgewandelten Inhalt, bis die vollständige Transformation beendet ist.

Inhaltspufferung wird für XSLT-Stylesheets benötigt, die Header manipulieren. Da Header vor dem Antworttext geschrieben werden, muss der Text gepuffert werden, damit die hinzugefügte Header-Information eingeschlossen werden kann.

Es gibt vier FileMaker-Erweiterungsfunktionen, für die das XSLT-Umwandlungsergebnis gepuffert werden muss:

- fmxslt:create_session(): Weitere Informationen finden Sie unter „Verwenden der Sitzungserweiterungsfunktionen“ auf Seite 65.
- fmxslt:set_header(): Weitere Informationen finden Sie unter „Verwenden von Header-Funktionen“ auf Seite 68.
- fmxslt:set_status_code (): Weitere Informationen finden Sie unter „Verwenden von Header-Funktionen“ auf Seite 68.
- fmxslt:set_cookie(): Weitere Informationen finden Sie unter „Verwenden der Cookie-Erweiterungsfunktionen“ auf Seite 69.

Damit diese FileMaker-Erweiterungsfunktionen korrekt arbeiten, müssen Sie die folgende XSLT-Verarbeitungsanweisung für die Abfrage in das Dokument der obersten Ebene aufnehmen:

```
<?xslt-cwp-buffer buffer-content="true"?>
```

Wichtig Wenn ein Basis-Stylesheet ein weiteres Stylesheet enthält, muss das Basis-Stylesheet die Verarbeitungsanweisung <?xslt-cwp-buffer?> enthalten. Diese Anweisung wird ignoriert, wenn sie in einem eingeschlossenen Stylesheet verwendet wird.

Ein Vorteil der Verwendung der Verarbeitungsanweisung zur Pufferung der Antwort ist, dass die Web Publishing Engine die Länge der Antwort ermitteln und den Content-Length-Header in der Antwort setzen kann. Durch Puffern der Antwort kann die Leistung der Web Publishing Engine beeinträchtigt werden.

Verwenden von Web Publishing Engine-Sitzungen zum Speichern von Informationen zwischen Abfragen

Mithilfe der Server-seitigen Sitzungen der Web Publishing Engine können Sie jeden Informationstyp zwischen Abfragen verfolgen und speichern. Sitzungen erlauben Ihnen, eine Web-Anwendung zu erstellen, die mithilfe persistenter beliebiger Informationen den Status zwischen Abfragen beibehalten kann. So könnten beispielsweise Benutzer-Client-Informationen, die auf einer ersten Formularseite eingegeben wurden, in einer Sitzung gespeichert und dann zum Ausfüllen der Werte auf einer späteren Seite verwendet werden.

Standardmäßig speichert die Web Publishing Engine die Sitzungs-ID in einem Cookie. Um Clients zu berücksichtigen, die keine Cookies erlauben, können Sie dem URL die Sitzungs-ID mithilfe der Funktion fmxslt:session_encode_url() hinzufügen. Damit in allen Situationen Kompatibilität gewährleistet ist, sollten Sie alle URLs auf der Seite mit der Funktion fmxslt:session_encode_url() kodieren. Diese Funktion fügt Ihrem URL den durch Semikolons getrennten Parameter jsessionid hinzu, der die Kennung für die übergeordnete Sitzung des jeweiligen Clients darstellt.

Anstatt den folgenden Link auf eine Seite zu setzen:

```
<a href="my_stylesheet.xsl?–db=products&–lay=sales&–grammar=fmresultset&–findall">hyperlinked text</a>
```

Sollten Sie alle Links wie folgt auf eine Seite platzieren:

```
<a href="{fmxslt:session_encode_url('my_stylesheet.xsl?–db=products&–lay=sales&–grammar=fmresultset
&–findall')}">hyperlinked text</a>
```

Wenn der Client keine Cookies zulässt, beinhaltet die Seite:

```
<a href="my_stylesheet.xsl;jsessionid=<session id>?–db=products&–lay=sales&–grammar=fmresultset&–findall">
hyperlinked text</a>
```

Wenn die Web Publishing Engine erkennt, dass der Client Cookies erlaubt, speichert die Funktion fmxslt:session_encode_url() die Sitzungs-ID in einem Cookie anstelle eines URLs.

Hinweis Sitzungsinformationen bleiben nach einem Neustart der Web Publishing Engine nicht erhalten.

Verwenden der Sitzungserweiterungsfunktionen

Verwenden Sie die folgenden Erweiterungsfunktionen für Sitzungen, um Sitzungsvariablen zu manipulieren. Sie können einen String, eine Zahl, einen booleschen Wert oder eine Knotengruppe (node-set) in einem Sitzungsobjekt speichern. Mithilfe einer Knotengruppe (node-set) können Sie eine Datenstruktur in XML anlegen und sie dann zwischen Abfragen im Sitzungsobjekt speichern.

Sitzungserweiterungsfunktion	Zurückgegebener Datentyp	Beschreibung
fmxslt:session_exists(String Sitzungsname)	Boolescher Wert	Prüft, ob eine Sitzung mit dem angegebenen Namen existiert.
fmxslt:create_session(String Sitzungsname)	Boolescher Wert	Erstellt eine Sitzung mit dem angegebenen Sitzungsnamen und der Standard-Zeitüberschreitung, die über Admin Console gesetzt wird. Weitere Informationen finden Sie in der FileMaker Server Hilfe. **Hinweis** Diese Funktion verlangt die Verarbeitungsanweisung <?xslt-cwp-buffer?>. Weitere Informationen finden Sie unter „Verwenden von Inhaltspufferung“ auf Seite 63.
fmxslt:invalidate_session(String Sitzungsname)	Boolescher Wert	Erzwingt eine sofortige Zeitüberschreitung der Sitzung.
fmxslt:set_session_timeout(String Sitzungsname, Number Zeitüberschreitung)	Boolescher Wert	Setzt die Sitzungszeitüberschreitung in Sekunden. Die Standardzeitüberschreitung für Sitzungen wird über Admin Console eingestellt.
fmxslt:session_encode_url(String URL)	String	Kodiert einen URL mit der Sitzungs-ID, wenn der Client keine Cookies unterstützt; gibt andernfalls den eingegebenen URL zurück.
fmxslt:set_session_object(String Sitzungsname, String Name, Object Wert)	XSLT-Objekt (Zahl, String, Boolescher Wert oder Knotengruppe)	Speichert ein XSLT-Objekt (Zahl, String, Boolescher Wert oder Knotengruppe (node-set)) unter einer Sitzung, die später über die Funktion fmxslt:get_session_object () abgerufen werden kann. Diese Funktion gibt auch das zuvor unter dem angegebenen Sitzungsobjektnamen gespeicherte Objekt zurück. Wenn kein Objekt unter dem Namen gespeichert wurde, wird ein Null-Objekt zurückgegeben. **Hinweis** Die Erweiterungsfunktion set_session_object() kann nur Zeichenfolgenwerte speichern und interpretiert an sie übergebene Objekte als Zeichenfolge. Wenn das Objekt nicht in eine Zeichenfolge konvertiert werden kann, ist kein Wert in der Sitzung gespeichert und der Fehlercode der Erweiterungsfunktion ist auf 10100 (Unbekannter Sitzungsfehler) gesetzt. Jeder Versuch, ein Sitzungsobjekt mit null oder einer leeren Zeichenfolge zu setzen, resultiert ebenfalls im Fehlercode 10100 (Unbekannter Sitzungsfehler). Um eine Sitzungsvariable zu löschen, entfernen Sie sie aus der Sitzung mithilfe der Funktion remove_session_object().
fmxslt:get_session_object(String Sitzungsname, String Name)	XSLT-Objekt	Bezieht ein XSLT-Objekt aus der Sitzung.
fmxslt:remove_session_object(String Sitzungsname, String Name)	XSLT-Objekt	Gibt ein XSLT-Objekt zurück und entfernt es dann aus der Sitzung.

Beispiel für das Erstellen einer Sitzung und anschließendes Speichern einer bevorzugten Farbe in der Sitzung:

```
<xsl:variable name="session">
    <xsl:choose>
        <xsl:when test="not (fmxslt:session_exists(string($session-name)))">
            <xsl:value-of select="fmxslt:create_session(string($session-name))"/>
        </xsl:when>
        <xsl:otherwise>true</xsl:otherwise>
    </xsl:choose>
</xsl:variable>
<xsl:variable name="favorite-color" select="fmxslt:set_session_object(string($session-name), 'favorite-color', string($color))"/>
```

Wichtig

- Um sicherzustellen, dass Benutzer nach Abschluss einer Sitzung von einer Datenbank abgemeldet werden, erzwingen Sie anhand der Funktion fmxslt:invalidate_session () eine sofortige Zeitüberschreitung der Sitzung.
- Wenn Sie Variablenfelder oder ein Script benutzen, das einen Status festlegt oder ändert, müssen Sie mithilfe von Admin Console die Option XSLT-Datenbanksitzungen für die Web Publishing Engine aktivieren. Andernfalls werden Variablenfeldwerte und Status zwischen Abfragen nicht beibehalten. Weitere Informationen finden Sie in der FileMaker Server Hilfe.
- Wenn Sie bei der Verwendung von Web Publishing Engine-Sitzungen in eine andere Datei wechseln, werden globale Feldwerte nicht beibehalten. Die Web Publishing Engine schließt die erste Datei, bevor sie die zweite Datei öffnet. Alternativ können Sie auf Daten aus der zweiten Datenbank zugreifen, indem Sie ein Layout in der ersten Datenbankdatei verwenden.

Senden von E-Mail-Nachrichten von der Web Publishing Engine

Mithilfe der Web Publishing Engine können Sie E-Mail-Nachrichten generieren, was für eigene Web-Lösungen nützlich ist. Damit die Web Publishing Engine eine E-Mail-Nachricht sendet, verwenden Sie eine der folgenden drei fmxslt:send_email ()-Erweiterungsfunktionen in einem XSLT-Stylesheet. Sie können diese Funktionen benutzen, um eine oder mehrere separate Nachrichten zu senden. Da die fmxslt:send_email ()-Funktionen im Server-seitigen XSLT-Stylesheet der Web Publishing Engine enthalten sind, kann ein Client keine unautorisierten E-Mail-Nachrichten über die Web Publishing Engine senden.

E-Mail-Erweiterungsfunktion	Zurückgegebener Datentyp	Beschreibung
fmxslt:send_email(String smtpFelder, String Nachrichtentext)	Boolescher Wert	Sendet eine E-Mail-Nachricht beliebiger Länge in Standardtext aus der Web Publishing Engine und verwendet dafür die Standard-Textkodierung der Web Publishing Engine.
fmxslt:send_email(String smtpFelder, String Nachrichtentext, String Kodierung)	Boolescher Wert	Sendet eine E-Mail-Nachricht beliebiger Länge in Standardtext und verwendet dafür eine der folgenden Textkodierungen: US-ASCII, ISO-8859-1, ISO-8859-15, ISO-2022-JP, Shift_JIS, UTF-8. Informationen über diese Kodierungen finden Sie unter „Festlegen von Textkodierung für Abfragen“ auf Seite 57.

E-Mail-Erweiterungsfunktion	Zurückgegebener Datentyp	Beschreibung
fmxslt:send_email(String smtpFelder, String xsltDatei, Node xml, boolean inklBilder)	Boolescher Wert	Sendet eine HTML-basierte E-Mail-Nachricht mit der Zeichenkodierung, die über das encoding-Attribut des <xsl:output>-Elements im Stylesheet angegeben ist. Wenn das encoding-Attribut nicht im <xsl:output>-Element angegeben wurde, wird für E-Mail-Nachrichten die Standard-Textkodierung der Web Publishing Engine verwendet.

Notizen

- In jeder der drei Formen der Funktion fmxslt:send_email () ist der Parameter smtpFields ein URL-kodierter String beliebiger Länge mit einer Adresse und Betreffdaten im folgenden Format, das auf RFC 2368, dem mailto-URL-Schema basiert:

  benutzername@host?name1=wert1&name2=wert2...

  Dabei gilt: benutzername@host gibt einen Empfänger an. Die Name-Wert-Paare können in beliebiger Reihenfolge angegeben werden und werden wie folgt definiert:

  - from=benutzername@host (darf nur einmal auftreten). Das from-Feld muss angegeben werden.
  - to=benutzername@host. Verwenden Sie dieses Name-Wert-Paar für zusätzliche Empfänger.
  - reply-to=benutzername@host (kann nur einmal vorhanden sein)
  - cc=benutzername@host
  - bcc=benutzername@host
  - subject=string (kann nur einmal vorhanden sein)

  Wenn die Felder from, reply-to oder subject öfter als einmal angegeben werden, wird die E-Mail-Nachricht nicht gesendet, die Funktion gibt den Wert false() zurück und der entsprechende Fehlerstatus wird gesetzt.
- Die Web Publishing Engine überprüft die Syntax aller angegebenen E-Mail-Adressen. Sie müssen die folgende Form haben:

  user@host.tld oder "genannter Kennzeichner"<user@host.tld>

  Dabei gilt: tld ist eine beliebige Domäne der oberen Ebene wie com oder net. Wenn beliebige dieser Felder eine ungültige E-Mail-Adresse enthalten, wird die E-Mail-Nachricht nicht gesendet und der entsprechende Fehlerstatuscode wird gesetzt.
- Die einzelnen Werte für den Parameter smtpFelder, z. B. der Betreff (subject), müssen URL-kodierte Strings sein. Beispiel: Das Zeichen & muss als & und Leerzeichen als %20 angegeben werden. Der vollständige String für den Parameter smtpFelder muss XML-kodiert sein (siehe das Beispiel am Ende dieses Abschnitts).
- Für jede dieser Funktionen wird der Wert true() zurückgegeben, wenn die E-Mail-Nachricht erfolgreich gesendet wurde; andernfalls wird false() zurückgegeben.
- Für E-Mail-Nachrichten verwendet die Web Publishing Engine anfangs die Textkodierung ISO-8859-1 als Standardeinstellung. Sie können diese Einstellung über Admin Console ändern. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

- Die Funktion fmxslt:send_email(String smtpFelder, String xsltDatei, Node xml, boolean inklBilder) sendet eine E-Mail-Nachricht bestehend aus XML-Daten, die durch das in dieser Funktion angegebene E-Mail-Stylesheet verarbeitet werden.
  - Geben Sie für den Parameter xsltDatei den Namen des E-Mail-Stylesheets an, indem Sie einen URL eingeben, der relativ zur Hauptverarbeitungs-Stylesheet-Datei für die Abfrage ist.
  - Geben Sie für den Parameter xml den übergeordneten Knoten der XML-Daten an, die Sie mit dem E-Mail-Stylesheet verwenden wollen. Um eine E-Mail-Nachricht mit denselben XML-Daten zu senden, die im Browser angezeigt werden, geben Sie einfach den XPath für das Root-Verzeichnis des Dokuments an: „/“. Andernfalls können Sie ein anderes XML-Dokument verwenden, indem Sie es zunächst mithilfe der Funktion document() laden und es dann an die Funktion fmxslt:send_email() übergeben.
  - Geben Sie für den Parameter inklBilder den booleschen Wert true() an, damit die Web Publishing Engine alle Bilder einschließt, die in der HTML der E-Mail-Nachricht als Anlagen angegeben sind. Dieser Parameter schließt sowohl die FileMaker-Datenbankbilder als auch andere Bilder von anderen Speicherorten ein. Die Web Publishing Engine ändert die Bild-URLs so, dass sie auf die Anlagen verweisen. Bei sehr zahlreichen oder großen Bildern kann die Leistung beeinträchtigt werden. Wenn Sie false () angeben, ändert die Web Publishing Engine die URLs für die Bilder nicht. Wenn die URLs absolut sind, versucht der E-Mail-Client, die Bilder vom Web-Server zu laden.

Beispiel für die Verwendung der Funktion fmxslt:send_email(String smtpFelder, String xsltDatei, Node xml, boolean inklBilder) in einer XPath-Anweisung, z. B. im <xsl:if>-Element:

```
fmxslt:send_email('tom_jones@company.com?subject=project%20status&amp;from=john_smith@company.com
&amp;cc=jane_doe@company.com','my_mail_template.xsl', /, true())
```

Informationen über die Konfiguration der Web Publishing Engine für die Verbindung zu einem SMTP-Server finden Sie in der FileMaker Server Hilfe.

Verwenden von Header-Funktionen

Sie können die Funktion fmxslt:get_header() verwenden, um Informationen aus der HTTP-Abfrage und Antwort-Headern zu lesen, und die Funktion fmxslt:set_header(), um Informationen in die Header zu schreiben. Diese Funktionen sind nützlich, wenn der Client mithilfe der Header-Informationen Daten aus dem Web-Server abrufen kann oder wenn Sie aus anderen Gründen einen HTTP-Header einrichten müssen.

Header-Erweiterungsfunktion	Zurückgegebener Datentyp	Beschreibung
fmxslt:get_header(String name)	String	Gibt den angegebenen Header-Wert zurück.
fmxslt:set_header(String name, String wert)	Ungültig	Legt den angegebenen Header-Wert fest.
fmxslt:set_status_code(Number status-code)	Ungültig	Setzt den HTTP-Statuscode.

Notizen

- Der in den Funktionen fmxslt:get_header() und fmxslt:set_header() verwendete Name sowie der in der Funktion fmxslt:set_header() verwendete Wert können ein String beliebiger Länge sein.
- Die Funktion fmxslt:set_header() und die Funktion fmxslt:set_status_code() verlangen die Verarbeitungsanweisung <?xslt-cwp-buffer?>. Weitere Informationen finden Sie unter „Verwenden von Inhaltspufferung“ auf Seite 63.

Die folgenden Beispiele demonstrieren, wie der Wert des Headers gesetzt wird. Nehmen Sie an, Sie verwenden ein Stylesheet zur Ausgabe einer vCard. Wenn ein Browser versucht, die Stylesheet-Seite zu laden, besteht potenziell das Problem, dass der Browser die .xsl-Datei als Stylesheet anstatt als vCard interpretieren könnte. Wenn Sie den Header „Inhalt-Disposition“ verwenden, können Sie angeben, dass eine Anlage mit der Erweiterung .vcf vorhanden ist.

```
<xsl:value-of select="fmxslt:set_header('Inhalt-Disposition','attachment;filename=test.vcf')" />
```

Verwenden der Cookie-Erweiterungsfunktionen

Mithilfe der Cookie-Erweiterungsfunktionen können Sie Cookies abrufen oder einrichten, die im Webbrowser des Clients gespeichert sind.

Cookie-Erweiterungsfunktion	Zurückgegebener Datentyp	Beschreibung
fmxslt:get_cookie(String name)	Knotengruppe	Gibt die COOKIE-Knotengruppe node-set zurück, die den angegebenen Cookie-Namen enthält.
fmxslt:get_cookies()	Knotengruppe	Gibt die COOKIE-Knotengruppe (node-set) mit allen vom Client gelieferten Cookies zurück.
fmxslt:set_cookie(String name, String wert)	Ungültig	Speichert das angegebene Cookie mit dem angegebenen Wert im Browser des Clients.
fmxslt:set_cookie(String name, String wert, Number ablauf, String pfad, String domäne)	Ungültig	Speichert das angegebene Cookie mit allen für ein Cookie verfügbaren Werten im Browser des Clients. Der Parameter ablauf gibt die Anzahl an Sekunden bis zum Ablauf des Cookies an.

Notizen

- Die Funktionen fmxslt:get_cookie() und fmxslt:get_cookies() geben eine Knotengruppe (node-set) mit der folgenden Struktur zurück:

```
<!ELEMENT cookies (cookie)*>
    <!ATTLIST cookie xmlns CDATA #FIXED "http://www.filemaker.com/xml/cookie">
<!ELEMENT cookie (#PCDATA)>
    <!ATTLIST cookie name CDATA #REQUIRED>
```

- Der XML-Namespace für die Cookies node-set ist "http://www.filemaker.com/xml/cookie". Stellen Sie sicher, dass Sie den Namespace deklarieren und einen Präfix für den Namespace angeben.
- Alle Parameterwerte für die fmxslt:set_cookie-Funktionen müssen gültig sein, da der Webbrowser sonst die fmxslt:set_cookie-Funktionsabfragen ignoriert.
- Für alle Cookie-Funktionen können die String-Parameter eine beliebige Länge aufweisen.
- Beide Formen der Funktion fmxslt:set_cookie() verlangen die Verarbeitungsanweisung <?xslt-cwp-buffer?>. Weitere Informationen finden Sie unter „Verwenden von Inhaltspufferung“ auf Seite 63.

Beispiel: get_cookie

Das folgende Beispiel ruft einen Cookie preferences und seinen Wert ab:

```
<xsl:variable name="pref_cookie" select="fmxslt:get_cookie('preferences')"/>
<xsl:value-of select="concat('Cookie Name = ', $pref_cookie/fmc:cookies/fmc:cookie/@name)"/> <br/>
<xsl:value-of select="concat('Cookie Value = ', $pref_cookie/fmc:cookies/fmc:cookie)"/>
```

Beispiel: set_cookie

Beispiel für das Setzen eines Cookies mit allen Werten:

```
<xsl:variable name="storing_cookie" select="fmxslt:set_cookie ('text1', 'text2', 1800, 'my_text', 'my.company.com')" />
```

Verwenden der Erweiterungsfunktionen zur Stringmanipulation

Sie können mithilfe der Funktionen zur Stringmanipulation die Kodierung von Strings beliebiger Länge ändern.

Erweiterungsfunktion zur Stringmanipulation	Zurück-gegebener Datentyp	Beschreibung
fmxslt:break_encode(String wert)	String	Gibt einen HTML-Break-kodierten String zurück. Zeichen wie & (kaufmännisches UND) werden durch & ersetzt. Zeilenumbruchzeichen wie Zeilenvorschub und Absatzende-Zeichen werden durch ersetzt. Diese Funktion ist nur wirksam, wenn das Attribut disable-output-escaping der Elemente <xsl:value-of> und <xsl:text> auf „yes“ gesetzt ist (disable-output-escaping="yes"). **Hinweis** Um ein Zeilenvorschub- oder Absatzende-Zeichen in den String einzufügen, auf den die Funktion fmxslt:break_encode() angewendet wird, müssen Sie die folgenden Escape-Zeichen im String verwenden: „
“ (für Zeilenvorschub) bzw.
“ (für Absatzende). Sie können kein Zeilenvorschub- oder Absatzende-Zeichen in den String einfügen, indem Sie in Ihrem Texteditor den Zeilenschalter drücken.
fmxslt:html_encode(String wert)	String	Gibt einen HTML-kodierten String zurück; Zeichen wie & (kaufmännisches UND) werden durch & ersetzt.
fmxslt:url_encode(String wert)	String	Gibt einen URL-kodierten String zurück. URL-Kodierung wird insbesondere für URLs verwendet, um Zeichen über das Internet zu übertragen. Zum Beispiel ist %26 die URL-kodierte Form von & (kaufmännisches Und). Wenn ein reserviertes Zeichen in Ihrem href verwendet wird, benutzen Sie diese Funktion zur URL-Kodierung Ihres Strings.
fmxslt:url_encode(String wert, String kodierung)	String	Gibt einen URL-kodierten String mit der Zeichenkodierung zurück, die Sie im Parameter kodierung angeben. Mögliche Kodierungen sind: US-ASCII, ISO-8859-1, ISO-8859-15, ISO-2022-JP, Shift_JIS und UTF-8. Verwenden Sie diese Funktion in Situationen, in denen Sie wissen, dass sich die von einem Web-Server erwartete Zeichenkodierung von derjenigen unterscheidet, die in Ihrem aktuellen Stylesheet verwendet wird. Beispiel: Ihre Website-Startseite kann in UTF-8 angezeigt sein, aber Benutzer klicken auf einen Link, um zu einer japanischen Seite zu wechseln. Wenn die Abfrage japanische Zeichen enthält und die japanischen Seiten Shift_JIS-Kodierung verwenden, wird der String am besten in Shift_JIS kodiert.
fmxslt:url_decode(String wert)	String	Gibt einen URL-dekodierten String zurück, der zuvor kodiert war.
fmxslt:url_decode(String wert, String kodierung)	String	Gibt einen URL-dekodierten String mit der Zeichenkodierung zurück, die Sie im Parameter kodierung angeben. Mögliche Kodierungen sind: US-ASCII, ISO-8859-1, ISO-8859-15, ISO-2022-JP, Shift_JIS und UTF-8. Verwenden Sie diese Funktion in Situationen, in denen Sie die in einem URL-kodierten String verwendete Zeichenkodierung angeben müssen, um den String korrekt zu dekodieren. Beispiel: Selbst wenn Ihre Website ISO-8859-1 verwendet, könnten Benutzer ein Formular mit einer anderen Zeichenkodierung senden.

Vergleich von Strings mithilfe regulärer Perl 5-Ausdrücke

Sie können die Erweiterungsfunktion fmxslt:regex_contains() verwenden, um Strings mithilfe regulärer Perl 5-Ausdrücke zu vergleichen. Ein Vergleich mit regulären Ausdrücken ist eine fortschrittliche Art von Textabgleich, bei der Sie bestimmen können, ob ein String einem bestimmten Muster entspricht. Syntax dieser Funktion:

fmxslt:regex_contains(String eingabe, String muster)

Dabei gilt: eingabe ist ein String und muster ist ein regulärer Perl 5-Ausdruck. Weitere Informationen zur Syntax von regelmäßigen Perl 5-Ausdrücken finden Sie unter www.perldoc.com. Die Funktion fmxslt:regex_contains() gibt einen booleschen Wert zurück.

Diese Funktion ist nützlich, wenn Sie anspruchsvollere Möglichkeiten zur Stringmanipulation benötigen, die von Standard-XSLT nicht zur Verfügung gestellt werden. Sie können z. B. feststellen, ob ein Feldwert eine gültige Telefonnummer oder E-Mail-Adresse enthält, indem Sie den String mit einem regulären Perl 5-Ausdruck vergleichen.

Beispiel für den Einsatz dieser Funktion, um zu ermitteln, ob ein Feldwert korrekt konstruierte E-Mail-Adressen enthält:

```
<xsl:variable name="email" select="'foo@bar.com'"/>
<xsl:if test="fmxslt:regex_contains($email,'^\w+[\w-\.]*\@\w+((-\w+)|(\w*))\.[a-z]{2,3}$')">Valid Email</xsl:if>
```

Wenn die Web Publishing Engine das Muster nicht analysieren kann, wird der Fehlerstatus auf Fehlercode 10311 gesetzt. Weitere Informationen finden Sie unter „Fehlercodenummern für die FileMaker XSLT-Erweiterungsfunktionen“ auf Seite 115.

Prüfen von Werten in einem Feld, das als Kontrollkästchen formatiert ist

Mithilfe der folgenden Erweiterungsfunktion können Sie ermitteln, ob ein bestimmter Wert in einer Kontrollkästchen-Werteliste in einem Feld in der FileMaker-Datenbank gespeichert ist:

fmxslt:contains_checkbox_value(String werteString, String werteListenEintrag)

Dabei gilt: werteString ist ein XPath, der das Feld angibt, und werteListenEintrag ist der Wert, den Sie suchen.

Wenn der angegebene Wert im Feld gespeichert ist, gibt diese boolesche Funktion true() zurück. Andernfalls gibt sie false() zurück. Sie können anhand dieser Funktion ermitteln, ob das geprüfte Attribut in einem HTML-Formular so eingestellt werden soll, dass es ein Kontrollkästchen als markiert zeigt.

Nehmen Sie z. B. an, ein Feld in einem FileMaker-Datenbanklayout hat die folgenden Kontrollkästchenoptionen:

[ ] Rot

[ ] Blau

[ ] Grün

[ ] Klein

[ ] Mittel

[ ] Groß

Wenn ein Benutzer nur Rot auswählt, würde das Feld den String „Rot“ enthalten. Anhand des folgenden Funktionsaufrufs könnten Sie feststellen, ob das Feld Blau enthält:

fmxslt:contains_checkbox_value(<feldwert_knoten>,'Blau')

Dabei gilt: <feldwert_knoten> ist der XPath zum <data>-Element für das Kontrollkästchenfeld. Die Funktion würde in diesem Beispiel false zurückgeben.

Eine übliche Anwendung dieser Funktion ist die Anzeige der Kontrollkästchen-Werteliste in einer Web-Seite und die Auswahl der Kontrollkästchen auf der Web-Seite, die auch in der Datenbank ausgewählt sind. Die folgenden HTML- und XSLT-Anweisungen erstellen eine Gruppe von Kontrollkästchen für ein Feld mit dem Namen style, das eine Werteliste mit dem Namen color_size benutzt:

```
<xsl:variable name="field-value" select="fmrs:field[@name='style']/fmrs:data" />
<xsl:for-each select="$valuelists[@NAME = 'color_size']/fml:VALUE">
        <input type="checkbox">
                <xsl:attribute name="name">style</xsl:attribute>
                <xsl:attribute name="value"><xsl:value-of select="."/></xsl:attribute>
                <xsl:if test="fmxslt:contains_checkbox_value($field-value,.)">
                        <xsl:attribute name="checked">checked</xsl:attribute>
                </xsl:if>
        </input><xsl:value-of select="."/><br/>
</xsl:for-each>
```

Die HTML- und XSLT-Anweisungen im Beispiel würden die folgenden Kontrollkästchen auf einer Web-Seite ausgeben, von denen Rot und Mittel ausgewählt sind:

[x] Rot

[ ] Blau

[ ] Grün

[ ] Klein

[x] Mittel

[ ] Groß

Verwenden der Datums-, Zeit- und Tages-Erweiterungsfunktionen

Mithilfe von Erweiterungsfunktionen können Sie die aktuellen Werte für Datum, Uhrzeit oder Tag abrufen und zwei Datums-, Zeit- oder Tagesangaben vergleichen.

Die Funktionen in der folgenden Tabelle verwenden die „fm“-Formate unabhängig von den regionalen Einstellungen. Die „fm“-Formate sind MM/tt/jjjj für Datum, HH:mm:ss für Zeit und MM/tt/jjjj HH:mm:ss für Zeitstempel.

Um ein anderes oder bevorzugtes Format für die Ausgabewerte zu verwenden, verwenden Sie Berechnungsfunktionen oder JavaScript.

Datums-, Zeit-, Tages-Erweiterungsfunktionen	Zurückgegebener Datentyp	Beschreibung
fmxslt:get_date()	String	Gibt das aktuelle Datum im „fm“-Format zurück.
fmxslt:get_date(String format)	String	Gibt das aktuelle Datum in dem Format zurück, das Sie angeben. Geben Sie den String „short“ oder „long“ für den Parameter format ein.
fmxslt:get_time()	String	Gibt die aktuelle Zeit im „fm“-Format zurück.

Datums-, Zeit-, Tages-Erweiterungsfunktionen	Zurückgegebener Datentyp	Beschreibung
fmxslt:get_time(String format)	String	Gibt die aktuelle Uhrzeit in dem Format zurück, das Sie angeben. Geben Sie den String „short“ oder „long“ für den Parameter format ein.
fmxslt:get_day()	String	Gibt den aktuellen Tag im Kurzformat zurück.
fmxslt:get_day(String format)	String	Gibt den aktuellen Tag in dem Format zurück, das Sie angeben. Geben Sie den String „short“ oder „long“ für den Parameter format ein.
fmxslt:get_fm_date_format()	String	Gibt den Formatierungsstring für das „fm“-Datumsformat zurück: „MM/tt/jjjj“
fmxslt:get_short_date_format()	String	Gibt den Formatierungsstring für das kurze Datumsformat zurück: „M/t/jj“
fmxslt:get_long_date_format()	String	Gibt den Formatierungsstring für das lange Datumsformat zurück: "MMM t, jjjj"
fmxslt:get_fm_date_format()	String	Gibt den Formatierungsstring für das „fm“-Zeitformat zurück: "HH:mm:ss "
fmxslt:get_fm_timestamp_format()	String	Gibt den Formatierungsstring für das „fm“-Zeitstempelformat zurück: "MM/tt/jjjj HH:mm:ss"
fmxslt:get_short_time_format()	String	Gibt den Formatierungsstring für das kurze Zeitformat zurück: "h:mm a"
fmxslt:get_long_time_format()	String	Gibt den Formatierungsstring für das lange Zeitformat zurück: "h:mm:ss a z"
fmxslt:get_short_day_format()	String	Gibt den Formatierungsstring für das kurze Tagesformat zurück: "EEE"
fmxslt:get_long_day_format()	String	Gibt den Formatierungsstring für das lange Tagesformat zurück: "EEEE"
fmxslt:compare_date(String datum1, String datum2)	Zahl	Diese Funktion vergleicht zwei Datumswerte. Sie gibt eine negative Zahl zurück, wenn datum1 vor datum2 liegt. Sie gibt eine positive Zahl zurück, wenn datum1 nach datum2 liegt. Sie gibt eine 0 zurück, wenn datum1 identisch mit datum2 ist. Beide Datumswerte müssen im „fm“-Format angegeben sein.
fmxslt:compare_time(String zeit1, String zeit2)	Zahl	Diese Funktion vergleicht zwei Zeitwerte. Sie gibt eine negative Zahl zurück, wenn zeit1 vor zeit2 liegt. Sie gibt eine positive Zahl zurück, wenn zeit1 nach zeit2 liegt. Sie gibt eine 0 zurück, wenn zeit1 identisch mit zeit2 ist. Beide Zeitwerte müssen im „fm“-Format angegeben sein.
fmxslt:compare_day(String tag1, String tag2)	Zahl	Diese Funktion vergleicht zwei Tageswerte. Sie gibt eine negative Zahl zurück, wenn tag1 vor tag2 liegt. Sie gibt eine positive Zahl zurück, wenn tag1 nach tag2 liegt. Sie gibt eine 0 zurück, wenn tag1 identisch mit tag2 ist. Beide Tage müssen im kurzen Tagesformat angegeben sein.

Die Funktionen in der folgenden Tabelle verwenden spezielle Formatstrings, die ein Datums- und Zeitformat angeben. Weitere Informationen finden Sie im nächsten Abschnitt, “Erläuterung von Datums- und Zeitformatstrings.”.

Datums-, Zeit-, Tages-Erweiterungsfunktionen	Zurückgegebener Datentyp	Beschreibung
fmxslt:get_datetime(String datumsFormat)	String	Gibt das aktuelle Datum und die Uhrzeit zurück und verwendet dafür die Datums- und Zeitformatstrings.
fmxslt:convert_datetime(String altesFormat, String neuesFormat, String datum)	String	Konvertiert das angegebene Datum aus altes Format in einen String in der Form von neuesFormat. Die Strings altesFormat und neuesFormat müssen mithilfe der Datums- und Zeitformatstrings angegeben werden.
fmxslt:compare_datetime(String datumsFormat1, String datumsFormat2, String datum1, String datum2)	Zahl	Diese Funktion vergleicht datum1 und datum2, indem sie die Datumsangaben gemäß den jeweiligen Datumsformaten dekodiert. Sie gibt eine negative Zahl zurück, wenn datum1 vor datum2 liegt. Sie gibt eine positive Zahl zurück, wenn datum1 nach datum2 liegt. Sie gibt eine 0 zurück, wenn datum1 identisch mit datum2 ist. Die Strings datumsFormat1 und datumsFormat2 müssen mithilfe der Datums- und Zeitformatstrings angegeben werden.

Erläuterung von Datums- und Zeitformatstrings

Die Datums- und Zeitformate werden durch Datums- und Zeitmusterstrings angegeben. Innerhalb von Datums- und Zeitmusterstrings werden Buchstaben ohne Anführungszeichen von A bis Z und von a bis z als Musterbuchstaben interpretiert, die Komponenten des Datums- oder Zeitstrings darstellen.

Die folgenden Musterbuchstaben sind definiert (alle anderen Zeichen von A bis Z und a bis z sind reserviert):

Buchstabe	Datums- oder Zeitkomponente	Darstellung	Beispiele
G	Ärabezeichnung	Text	n. Chr.
j	Jahr	Jahr	1996; 96
M	Monat im Jahr	Monat	Juli;Jul;07
w	Woche im Jahr	Zahl	27
W	Woche im Monat	Zahl	2
T	Tag im Jahr	Zahl	189
t	Tag im Monat	Zahl	10
F	Tag der Woche im Monat	Zahl	2
E	Wochentag	Text	Mittwoch;Mi
a	AM/PM-Markierung	Text	Nachmittag
H	Stunde des Tages (0-23)	Zahl	0
k	Stunde des Tages (1-24)	Zahl	24
K	Stunde vormittags/nachmittags (0-11)	Zahl	0
h	Stunde vormittags/nachmittags (1-12)	Zahl	12

Buchstabe	Datums- oder Zeitkomponente	Darstellung	Beispiele
m	Minute in Stunde	Zahl	30
s	Sekunde in Minute	Zahl	55
S	Millisekunde	Zahl	978
z	Zeitzone	Allgemeine Zeitzone	Pazifische Sommerzeit;PDT
Z	Zeitzone	RFC 822 Zeitzone	-0800

Musterbuchstaben werden gewöhnlich wiederholt, da ihre Anzahl die exakte Darstellung bestimmt:

- **Text**: Wenn die Anzahl der Musterbuchstaben vier oder mehr beträgt, wird für die Formatierung die ausführliche Form verwendet. Andernfalls wird eine kurze oder abgekürzte Form verwendet, falls verfügbar. Beim Parsing werden beide Formen unabhängig von der Anzahl der Musterbuchstaben berücksichtigt.
- **Zahl**: Beim Formatieren gibt die Anzahl der Musterbuchstaben die minimale Stellenanzahl an, Zahlen mit weniger Stellen werden bis zu dieser Anzahl mit Nullen aufgefüllt. Beim Parsing wird die Anzahl der Musterbuchstaben ignoriert, sofern sie nicht zur Trennung zweier benachbarter Felder erforderlich sind.
- **Jahr**: Wenn beim Formatieren zwei Musterbuchstaben vorhanden sind, wird das Jahr auf zwei Stellen abgeschnitten. Andernfalls wird es als eine Zahl interpretiert.

  Wenn beim Parsing mehr als zwei Musterbuchstaben vorhanden sind, wird das Jahr unabhängig von der Stellenanzahl buchstäblich interpretiert. Bei Verwenden des Musters MM/tt/jjjj wird die Angabe „11.01.12“ als 11. Jan 12 A.D. analysiert.

  Beim Parsing mit dem abgekürzten Jahresmuster (y oder yy) muss das abgekürzte Jahr in Relation zu einem Jahrhundert interpretiert werden, indem Datumsangaben so angepasst werden, dass sie im Zeitraum von 80 Jahren vor und 20 Jahren nach der Erstellung der Instanz liegen. Wenn Sie z. B. das Muster MM/tt/jj verwenden und eine Instanz mit diesem Datumsformat am 1. Januar 1997 erstellt wurde, wird der String „11.01.12“ als 11. Januar 2012 interpretiert. Dagegen wird der String „04.05.64“ als 4. Mai 1964 interpretiert. Beim Parsing werden nur Strings mit exakt zwei Stellen als das Standardjahrhundert analysiert. Jeder andere numerische String, z. B. ein einstelliger String, ein drei- oder mehrstelliger String oder ein zweistelliger String, der nicht ausschließlich aus Ziffern besteht (z. B. -1), wird buchstäblich interpretiert. Also werden 02.01.3 oder 02.01.003 mit demselben Muster wie 2. Jan 3 AD analysiert. Ebenso wird 02.01.-3 als 2. Jan 4 v. Chr. interpretiert.
- **Monat**: Wenn zwei oder mehr Musterbuchstaben vorhanden sind, wird der Monat als Text interpretiert, andernfalls als Zahl.
- **Allgemeine Zeitzone**: Zeitzonen werden als Text interpretiert, wenn sie Namen haben. Für Zeitzonen, die einen von GMT versetzten Wert haben, wird folgende Syntax verwendet:
  - GMTVersatzZeitzone. *GMT Vorzeichen Stunden:Minuten*
  - Vorzeichen. + oder -
  - Stunden. *Ziffer* oder *Ziffer Ziffer*
  - Minuten. *Ziffer Ziffer*
  - Ziffer. Eines von: 0 1 2 3 4 5 6 7 8 9

  Stunden müssen zwischen 0 und 23 liegen, *Minuten* zwischen 00 und 59. Das Format ist unabhängig von der Region und die Ziffern müssen dem Basic-Latin-Block des Unicode-Standards entnommen werden.

  Beim Parsing werden auch RFC 822-Zeitzonen akzeptiert.

- **RFC 822-Zeitzone**: Beim Formatieren wird das vierstellige RFC 822-Zeitzonenformat verwendet:
  - RFC822Zeitzone. *Vorzeichen ZweiStelligeStunden Minuten*
  - ZweiStelligeStunden. *Ziffer Ziffer*

  ZweiStelligeStunden muss zwischen 00 und 23 liegen. Andere Definitionen sind wie für allgemeine Zeitzonen.

  Beim Parsing werden auch allgemeine Zeitzonen akzeptiert.

Die folgenden Beispiele zeigen, wie Datums- und Zeitmuster in den USA interpretiert werden. Die angegebenen Datums- und Zeitwerte sind 2001-07-04 12:08:56 Ortszeit in der US-amerikanischen Zeitzone „Pacific Time“.

Datums- und Zeitmuster	Ergebnis
"jjjj.MM.tt G 'at' HH:mm:ss z"	2001.07.04 n. Chr. at 12:08:56 PDT
"EEE, MMM t, ' 'jj"	Mi, Jul 4, '01
"hh:mm" a	12:08 PM
"hh 'o' 'clock' a, zzzz"	12 o'clock PM, Pazifische Sommerzeite
"K:mm a, z"	0:08 PM, PDT
"jjjj.MMMMM.tt GGG hh:mm aaa"	02001.Juli.04 n. Chr. 12:08 PM
"EEE, t MMM jjjj HH:mm:ss Z"	Mi, 4 Jul 2001 12:08:56 -0700
"jjMMttHHmmssZ"	010704120856-0700

Copyright 2003 Sun Microsystems, Inc. Genehmigter Nachdruck.

Prüfen des Fehlerstatus von Erweiterungsfunktionen

Sie können mithilfe der folgenden Erweiterungsfunktion in einem XSLT-Stylesheet den aktuellen Fehlerstatus der zuletzt aufgerufenen FileMaker XSLT-Erweiterungsfunktion prüfen und Fehler behandeln, die bei der Verarbeitung Ihrer Seiten auftreten:

fmxslt:check_error_status()

Beim Aufruf der Funktion fmxslt:check_error_status() gibt die Web Publishing Engine den aktuellen Fehlercode für die zuletzt aufgerufene Funktion im Typ Zahl zurück und setzt den Fehlerstatus dann auf 0 („Kein Fehler“) zurück. Informationen über die Fehlercodewerte finden Sie unter „Fehlercodenummern für die FileMaker XSLT-Erweiterungsfunktionen“ auf Seite 115.

Verwenden der Protokollierung

Sie können mithilfe des XSLT-Standardelements <xsl:message> Protokolleinträge in die Protokolldatei der Web Publishing Engine schreiben. Weitere Informationen finden Sie unter „Verwenden des Web Publishing Engine-Programmprotokolls“ auf Seite 84.

Verwenden der Server-seitigen Verarbeitung von Scriptingsprachen

Der zugrunde liegende XSLT-Transformer, der in die Web Publishing Engine integriert ist, unterstützt die Server-seitige Verarbeitung von Scriptingsprachen. Daher können Sie JavaScript verwenden, um Ihre eigenen Erweiterungsfunktionen zu entwickeln, die direkt aus einem XSLT-Stylesheet aufgerufen werden können.

Zwei Java-Librarys werden installiert, um diese Funktionalität zu ermöglichen:

- bsf.jar – Diese Library ermöglicht dem XSLT-Transformer, Verbindungen zu Scriptingsprachen herzustellen.
- js.jsr – Diese Library ist eine vollständige JavaScript-Implementierung des Mozilla-Projekts.

Mit diesen Librarys können Sie Ihre eigenen Erweiterungsfunktionen innerhalb Ihres XSLT-Stylesheet-Codes erstellen. Diese Erweiterungsfunktionen können jede Scriptinglogik implementieren und sind besser zu verwalten, als XSLT und XPath zu verwenden, um logische Funktionen umzusetzen.

Weitere Informationen über die Erweiterungsunterstützung des XSLT-Transformers finden Sie auf der Apache Xalan Extensions-Website:

http://xml.apache.org/xalan-j/extensions.html

Definieren einer Erweiterungsfunktion

So definieren Sie eine Erweiterungsfunktion innerhalb Ihres Stylesheets:

1. Definieren Sie den Namespace für die Erweiterung.

   Fügen Sie den Namespace xalan hinzu, um den XSLT-Transformer anzuweisen, Erweiterungskomponenten zu unterstützen, indem Sie den Namen Ihres eigenen Erweiterungsfunktion-Namespace angeben. Das folgende Beispiel verwendet fmp-ex als Namespace-Präfix der Erweiterungsfunktion.

   ```
   <xsl:stylesheet version="1.0"
   xmlns:xsl=http://www.w3.org/1999/XSL/Transform
   xmlns:xalan=http://xml.apache.org/xslt
   xmlns:fmp-ex="ext1"
   exclude-result-prefixes="xsl xalan fmp-ex">
   ```

2. Definieren Sie die Erweiterungskomponente und die Erweiterungsfunktionen mit dem Code, der tatsächlich Ihre Erweiterungsfunktion implementiert.

   ```
   <xalan:component prefix="fmp-ex" functions="getValueColor">
   <xalan:script lang="javascript">
         function getValueColor(value) {
         if (value > 0)
               return ("#009900");
         else
               return ("#CC0000");
         }
   </xalan:script>
   </xalan:component>
   ```

   Dieses Beispiel gibt einen Farbwert basierend auf einem Eingabewert zurück. Wenn der Eingabewert größer als 0 ist, ist die zurückgegebene Farbe Grün ("#009900"). Wenn andernfalls der Wert kleiner 0 ist, ist die zurückgegebene Farbe Rot ("#CC0000").

   Hinweis Das Element <xalan:component> muss dem Element <xsl:stylesheet> untergeordnet sein.

3. Verwenden Sie die Erweiterungsfunktion innerhalb des Stylesheets.

Das folgende Beispiel zeigt, wie Sie eine Erweiterungsfunktion mittels einer Xpath-Anweisung aufrufen.

Das erste Beispiel würde die Schriftfarbe auf Grün ("#009900") setzen.

```
<font color="{fmp-ex:getValueColor(50)}">Der Wert ist 50</font>
```

Das zweite Beispiel würde die Schriftfarbe auf Rot ("##CC0000") setzen.

```
<font color="{fmp-ex:getValueColor(-500)}">Der Wert ist -500</font>
```

Ein Beispiel für eine Erweiterungsfunktion

Die einfache JavaScript-Funktion, die im obigen Prozess verwendet wird, könnte über eine <xsl:choose>-Anweisung implementiert worden sein. Aber der Kniff bei der Verwendung einer Scriptingerweiterung liegt darin, dass Sie eine Funktion erstellen können, die nicht alleine in XSLT oder XPath implementiert werden kann.

Nehmen wir zum Beispiel an, dass Sie ein Intranet-Portal für Ihr Unternehmen aufsetzen und Sie die aktuellen Aktiennotierungen auf dieser Portalseite aufnehmen möchten. Es gibt zwar XML-Börsen-Feeds, im Allgemeinen sind jedoch kommerzielle Lizenzen erforderlich, um auf sie zuzugreifen. Sie können jedoch Börsendaten in einem CSV-Dokument von Yahoo herunterladen. Die XPath document()-Funktion kann Inhalt aus XML-Quellen importieren, aber Sie müssten den CSV-Inhalt zu XML konvertieren. Eine Lösung besteht darin, dass Sie JavaScript verwenden, um die CSV-Börsendaten herunterzuladen, die Datei zu parsen und die Daten zu extrahieren.

Dieser URL zeigt die Syntax für den Abruf einer Börsennotierung aus Yahoo in Form einer CSV-Datei:

```
http://quote.yahoo.com/d/quotes.csv?s=<ticker>&f=l1gh&e=.csv
```

Dabei gilt: <ticker> steht für das Börsensymbol, für das Sie Daten abrufen möchten.

Die zurückgegebenen Daten bestehen aus drei Zahlen, die durch Kommata getrennt sind. Beispiel:

```
31.79,31.17,32.12
```

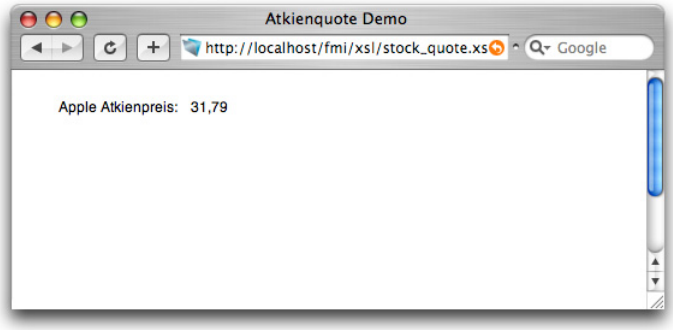
Der erste Wert ist der letzte Handelspreis, der zweite Wert das Tagestief und der dritte Wert das Tageshoch.

Das folgende Beispiel zeigt eine JavaScript-XSLT-Erweiterungsfunktion, die einen aktuellen Börsenkurs von Yahoo abruft und ihn über eine XPath-Funktion zur Verfügung stellt:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
        exclude-result-prefixes="xsl fmxslt fmrs xalan fmp-ex"
        version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:fmrs="http://www.filemaker.com/xml/fmresultset"
        xmlns:fmxslt="xalan://com.fmi.xslt.ExtensionFunctions"
        xmlns:xalan="http://xml.apache.org/xslt"
        xmlns:fmp-ex="ext1"
>

<?xslt-cwp-query params="-grammar=fmresultset&-process" ?>
<xsl:output method="html"/>
<xalan:component prefix="fmp-ex" functions="include get_quote" >
```

```
<xalan:script lang="javascript">
	function include(url) {
		var dest = new java.net.URL(url);
		var dis = new java.io.DataInputStream(dest.openStream());
		var res = "";
		while ((line = dis.readLine()) != null)
		{
			res += line + java.lang.System.getProperty("line.separator");
		}
		dis.close();
		return res;
	}
	function get_quote(ticker) {
		url = "http://quote.yahoo.com/d/quotes.csv?s="+
			"+ticker+"&amp;f=l1gh&amp;e=.csv";
		csv_file = include(url);
		var str_tokenizer = new java.util.StringTokenizer(csv_file, ',');
		// Der erste Token ist der letzte Handelspreis.
		var last = str_tokenizer.nextToken();
		return last;
	}
</xalan:script>
</xalan:component>

<xsl:template match="/fmrs:fmresultset">
	<html>
		<Datenbereich>
		<font size="2" face="verdana, arial">
		Apple Stock Price: <xsl:value-of select="fmp-ex:get_quote('AAPL')"/>
		</font>
		</body>
	</html>
</xsl:template>

</xsl:stylesheet>
```

Wenn die Web Publishing Engine dieses Stylesheet verarbeitet, ruft es die Börsendaten von Yahoo ab. Die Funktion get_quote() geht durch die Börsenkursdaten und gibt die Daten an das Stylesheet zurück. Die transformierte Ausgabe wird im Browser angezeigt:



Kapitel 7
Einstellen, Testen und Überwachen einer Site

Dieses Kapitel enthält Informationen über das Einstellen und Testen einer Custom Web Publishing-Site vor ihrem Einsatz in einer Produktionsumgebung. Zudem finden Sie hier Informationen zur Verwendung von Protokolldateien, um die Site während der Tests und nach der Freigabe zu überwachen.

Einstellen einer Custom Web Publishing-Site

Bevor Sie Ihre Site richtig testen können, müssen Sie die erforderlichen Dateien an die richtigen Speicherorte auf dem bzw. den Servern kopieren oder verschieben.

So stellen Sie Ihre Site zum Testen ein und bereiten Sie dafür vor:

1. Folgen Sie den Anleitungen in Kapitel 3, „Vorbereiten von Datenbanken für Custom Web Publishing“.
2. Stellen Sie sicher, dass XSLT und XML in FileMaker Server Admin Console aktiviert und richtig konfiguriert wurden.

   Hinweis Weitere Informationen finden Sie in der FileMaker Server Hilfe.
3. Stellen Sie sicher, dass der Web-Server und die Web Publishing Engine laufen.
4. Kopieren oder verschieben Sie die XSLT-Stylesheets auf den Rechner, auf dem sich die Web Publishing Engine befindet.

   Kopieren oder verschieben Sie die XSLT-Stylesheets in das folgende Verzeichnis auf dem Web Publishing Engine-Rechner:

   - Apache (Mac OS): /Library/FileMaker Server 9/Web Publishing/xslt-template-files
   - IIS (Windows): <Laufwerk>:\Programme\FileMaker\FileMaker Server 9\Web Publishing\xslt-template-files

   wobei <Laufwerk> das Hauptlaufwerk ist, von dem das System startet.

   Hinweis Sie können die Stylesheets auch in einem optionalen Ordner oder einer Ordnerhierarchie innerhalb des Ordners xslt-template-files platzieren.
5. Kopieren oder verschieben Sie referenzierte Medienobjekte auf den Web-Server-Rechner.

   Wenn ein Datenbank-Medienfeld einen Dateiverweis anstelle der eigentlichen Datei speichert, muss das referenzierte Medienobjekt im FileMaker Pro-Web-Ordner gespeichert sein, wenn der Datensatz erstellt oder bearbeitet wird. Um Ihre Site einzustellen, kopieren oder verschieben Sie die referenzierten Medien in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software.

   Hinweis Falls die Medienfelder die tatsächlichen Dateien in der FileMaker-Datenbank speichern, ist mit dem Inhalt der Medienfelder keine Aktion erforderlich, wenn die Datenbank richtig auf der Datenbank-Server-Komponente des FileMaker Server-Einsatzes bereitgestellt und zugänglich ist.
6. Verwenden Sie folgende URL-Syntax, um ein XSLT-Stylesheet anzufordern und zu verarbeiten und das resultierende HTML zu generieren.

   <Schema>://<Host>[:<Port>]/fmi/xsl/<Pfad>/<Stylesheet>.xsl[?<Query-String>]

   Dabei gilt:

   - <Schema> ist das HTTP- oder HTTPS-Protokoll.
   - <Host> ist die IP-Adresse oder der Domänenname des Host-Computers, auf dem der Web-Server installiert ist.

- <port> ist optional und gibt den Port an, den der Web-Server überwacht. Wird kein Port angegeben, wird der Standard-Port des Protokolls (Port 80 für HTTP oder Port 443 für HTTPS) angenommen.
- <path> ist optional und gibt den/die Ordner innerhalb des Ordners xslt-template-files an, in dem sich das XSLT-Stylesheet befindet.
- <Stylesheet> ist der Name des Stylesheets mit einer .xsl-Erweiterung.
- <query_string> kann eine Kombination aus einem Query-Befehl und einem oder mehreren Query-Parametern für Custom Web Publishing mit XSLT sein.

Wenn das angegebene Stylesheet eine <?xslt-cwp-query ?>-Verarbeitungsanweisung enthält, haben der statisch definierte Query-Befehl und die Parameter Vorrang vor anderen Query-Befehlen oder Parametern in der URL-Query-Zeichenfolge. Wenn Sie auf das Stylesheet home.xsl verweisen, das durch den XSLT-Site-Assistenten generiert wurde, müssen Sie keinen Query-String integrieren. Weitere Informationen zu Query-Strings finden Sie in Anhang A, „Gültige Namen in Query-Strings“..

Die URL-Syntax einschließlich der Namen der Query-Befehle und -Parameter unterscheidet mit Ausnahme von Teilen der Query-Zeichenfolge zwischen Groß- und Kleinschreibung. Der Großteil des URLs wird in Kleinbuchstaben angegeben. Wenn Sie z. B. Ihre Stylesheets (einschließlich eines home.xsl-Stylesheets) in den Ordner my_templates im Ordner xslt-template-files kopiert haben, können Sie folgenden URL verwenden, um die Stylesheets anzufordern und zu verarbeiten:

http://192.168.123.101/fmi/xsl/my_templates/home.xsl

Hinweis Die Web Publishing Engine erlaubt Web-Benutzern nicht, den Quelltext der XSLT-Stylesheets anzuzeigen, die im Ordner xslt-template-files installiert sind. Wenn ein Web-Benutzer eine Anforderung für die Verarbeitung eines Stylesheets sendet, sendet die Web Publishing Engine nur die HTML-Seiten an den Webbrowser, die das Ergebnis der XSLT-Site-Assistent-Stylesheets sind.

Testen einer Custom Web Publishing-Site

Bevor Sie Benutzern mitteilen, dass Ihre Custom Web Publishing-Site verfügbar ist, sollten Sie überprüfen, ob Aussehen und Funktionsweise Ihren Vorstellungen entsprechen.

- Testen Sie Funktionen wie das Suchen, Hinzufügen, Löschen und Sortieren von Datensätzen mit unterschiedlichen Konten und Zugriffsrechten.
- Überprüfen Sie, ob sich die Berechtigungen wie erwartet verhalten, indem Sie sich mit verschiedenen Konten anmelden. Stellen Sie sicher, dass nicht autorisierte Benutzer nicht auf Ihre Daten zugreifen oder sie ändern können.
- Überprüfen Sie alle Scripts, um sicherzustellen, dass das Ergebnis wie erwartet ist (siehe „FileMaker-Scripts und Custom Web Publishing“ auf Seite 22 für Informationen zur Gestaltung Web-freundlicher Scripts).
- Testen Sie Ihre Site mit unterschiedlichen Betriebssystemen und Webbrowsern.

Hinweis Wenn Sie über keine Netzwerkverbindung verfügen und FileMaker Server 9 – also Web-Server, Web Publishing Engine und Datenbank-Server – auf einem einzigen Computer installiert haben, können Sie Ihre Custom Web Publishing-Site testen, indem Sie im URL Folgendes verwenden: http://localhost/ oder http://127.0.0.1/. Informationen über die URL-Syntax finden Sie unter „Erläuterung der URL-Syntax für XML-Daten und Medienobjekte“ auf Seite 27, und „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“ auf Seite 52.

Beispiele für Stylesheets zum Testen der XML-Ausgabe

Hier sind zwei Beispiele für XSLT-Stylesheets, die sich zum Testen von XML-Ausgabe eignen.

- Das folgende Stylesheet-Beispiel gibt die angeforderten XML-Daten ohne Umwandlung aus. Dieses Stylesheet ist nützlich für die Anzeige der tatsächlichen XML-Daten, die die Web Publishing Engine verwendet.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:fmrs="http://www.filemaker.com/xml/fmresultset">
    <xsl:output method="xml"/>
    <xsl:template match="/">
        <xsl:copy-of select="."/>
    </xsl:template>
</xsl:stylesheet>
```

- Bei der Fehlerbehebung in einem Stylesheet können Sie das folgende Beispiel des HTML-Tags <textarea> verwenden, um das XML-Quelldokument anzuzeigen, auf das über das Stylesheet in einem abrollbaren Textbereich zugegriffen wurde. Auf derselben Seite können Sie die umgewandelten XSLT-Ergebnisse mit dem XML-Quelldokument vor der Umwandlung vergleichen.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:fmrs="http://www.filemaker.com/xml/fmresultset">
    <xsl:output method="html"/>
<html>
    <body>
    <xsl:template match="/fmrs:fmresultset">
        <textarea rows="20" cols="100">
            <xsl:copy-of select="."/>
        </textarea><br/>
    </xsl:template>
    </body>
</html>
</xsl:stylesheet>
```

Überwachen Ihrer Website

Sie können anhand der folgenden Typen von Protokolldateien Ihre Custom Web Publishing-Site überwachen und Informationen über Web-Benutzer sammeln, die Ihre Site besuchen:

- Web-Server-Zugriff und Fehlerprotokolle
- Web Publishing Engine-Programmprotokoll
- Web-Server-Modul-Fehlerprotokoll
- interne Web Publishing Core-Zugriffsprotokolle

Verwenden der Web-Server-Zugriffs- und Fehlerprotokolle

Apache (Mac OS): Der Apache-Web-Server generiert eine Zugriffsprotokolldatei und eine Fehlerprotokolldatei. Die Apache-Zugriffsprotokolldatei, standardmäßig im W3C Common Logfile Format, ist eine Aufzeichnung aller beim Web-Server eingehenden HTTP-Abfragen. Das Apache-Fehlerprotokoll ist eine Aufzeichnung von Problemen bei der Verarbeitung von HTTP-Abfragen. Weitere Informationen über diese Protokolldateien finden Sie in der Dokumentation zu Ihrem Apache-Web-Server.

IIS (Windows): Der Microsoft IIS-Web-Server generiert eine Zugriffsprotokolldatei und zeigt Fehler in der Windows-Ereignisanzeige, anstatt sie in eine Protokolldatei zu schreiben. Die Zugriffsprotokolldatei, standardmäßig im W3C Extended Log File Format, ist eine Aufzeichnung aller beim Web-Server eingehenden HTTP-Abfragen. Sie können auch das W3C Common Logfile Format für das Zugriffsprotokoll verwenden. Weitere Informationen finden Sie in der Dokumentation zum Microsoft IIS-Web-Server.

Informationen über das W3C Common Logfile Format und das W3C Extended Log File Format finden Sie auf der Website des World Wide Web Consortium: www.w3.org.

Verwenden des Web Publishing Engine-Programmprotokolls

Standardmäßig generiert die Web Publishing Engine eine Programmprotokolldatei, die eine Aufzeichnung von Web Publishing Engine-Fehler-, Script- und Benutzerprotokollinformationen darstellt.

- Die Fehlerprotokollinformation beschreibt alle ungewöhnlichen Web Publishing Engine-Fehler, die aufgetreten sind. Dem Web-Benutzer häufig gemeldete Fehler wie „Datenbank nicht geöffnet“ werden nicht aufgezeichnet.
- Die Scriptprotokollinformation beschreibt alle Fehler, die generiert wurden, während Web-Benutzer Scripts ausführten. Sie listet z. B. Scriptschritte auf, die übersprungen wurden, weil sie nicht Web-kompatibel sind.
- Die Benutzerprotokollmeldungen umfassen Meldungen, die vom XSLT-Element <xsl:message> in XSLT-Stylesheets erzeugt wurden. Immer wenn Web-Benutzer auf Ihr XSLT-Stylesheet zugreifen, werden in der Programmprotokolldatei Informationen aufgezeichnet, die Sie in einem <xsl:message>-Element angegeben haben. Weitere Informationen finden Sie unter Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Das Programmprotokoll heißt pe_application_log.txt und befindet sich im Ordner Logs, der sich im FileMaker Server-Ordner auf dem Web Publishing Engine-Host befindet.

Die Datei pe_application_log.txt wird generiert, wenn beliebige der folgenden Protokollierungsoptionen in der Web Publishing Engine aktiviert sind:

Protokollierungsoption aktiviert	In pe_application_log.txt aufgezeichnete Informationen
Fehlerprotokollierung	Ungewöhnliche Web Publishing Engine-Fehler, die aufgetreten sind. Dem Web-Benutzer häufig gemeldete Fehler wie „Datenbank nicht geöffnet“ werden nicht aufgezeichnet.
Scriptprotokollierung	Fehler, die erzeugt werden, wenn Web-Benutzer Scripts ausführen. Sie listet z. B. Scriptschritte auf, die übersprungen wurden, weil sie nicht Web-kompatibel sind.
Benutzerprotokollierung	Meldungen, die erzeugt werden, wenn Web-Benutzer auf Ihre Custom Web Publishing-Lösung zugreifen.

Alle drei Protokollierungsoptionen sind standardmäßig aktiviert. Informationen zur Einstellung dieser Optionen über die Admin Console finden Sie in der FileMaker Server Hilfe.

Hinweis Die Einträge im Programmprotokoll werden nicht automatisch gelöscht, daher kann die Datei mit der Zeit sehr umfangreich werden. Um Festplattenspeicherplatz auf Ihrem Host-Computer zu sparen, empfiehlt sich ein regelmäßiges Archivieren der Programmprotokolldatei.

Verwenden des Web-Server-Modul-Fehlerprotokolls

Wenn der Web-Server keine Verbindung zur Web Publishing Engine aufbauen kann, generiert das Web Server Modul eine Protokolldatei, die alle Fehler in Zusammenhang mit ihrer Ausführung aufzeichnet. Diese Datei heißt web_server_module_log.txt und befindet sich im Ordner Logs im FileMaker Server-Ordner auf dem Web-Server-Host.

Verwenden von internen Web Publishing Core-Zugriffsprotokollen

Die Softwarekomponente Web Publishing Core der Web Publishing Engine generiert zwei interne Zugriffsprotokolldateien, die jeden Zugriff auf den Web Publishing Core aufzeichnen.

- Das Zugriffsprotokoll wpc_access_log.txt enthält eine Aufzeichnung aller Endanwenderabfragen zur Generierung von XML und Verwendung von FileMaker Server Instant Web Publishing. Diese Abfragen werden vom Web-Server direkt an den Web Publishing Core geleitet.
- Das Zugriffsprotokoll pe_internal_access_log.txt enthält eine Aufzeichnung aller internen XML-Abfragen, die die Softwarekomponente XSLT-CWP der Web Publishing Engine ausführt, während XSLT-Abfragen verarbeitet werden. Diese Abfragen werden intern in der Web Publishing Engine von der Softwarekomponente XSLT-CWP an die Softwarekomponente Web Publishing Core geleitet.

Diese Protokolldateien befinden sich im Ordner Logs, der sich im FileMaker Server-Ordner auf dem Web Publishing Engine-Host befindet.

Die internen Zugriffsprotokolle werden generiert, wenn die Option Anmeldungsprotokollierung in der Web Publishing Engine aktiviert ist. Die Standardeinstellung ist aktiviert. Informationen zur Einstellung der Option Anmeldungsprotokollierung in Admin Console finden Sie in der FileMaker Server Hilfe.

Anhang A
Gültige Namen in Query-Strings

Dieser Anhang beschreibt die gültigen Namen für Query-Befehle und deren Parameter, die Sie in einem Query-String verwenden können, wenn Sie über die Web Publishing Engine auf FileMaker-Daten zugreifen.

Erläuterung von Query-Befehlen und -Parametern

Die folgende Liste enthält sämtliche Namen für Query-Befehle und Query-Parameter:

Query-Befehlsnamen	Query-Parameternamen
–dbnames (Siehe Seite 92.)	–db (Siehe Seite 95.)
–delete (Siehe Seite 92.)	–encoding (nur XSLT) (Siehe Seite 95.)
–dup (Siehe Seite 92.)	–field (Siehe Seite 95.)
–edit (Siehe Seite 92.)	fieldname (Siehe Seite 96.)
–find, –findall, –findany (Siehe Seite 92.)	fieldname.op (Siehe Seite 96.)
–findquery (Siehe Seite 93.)	–grammar (nur XSLT) (Siehe Seite 97.)
–layoutnames (Siehe Seite 93.)	–lay (Siehe Seite 97.)
–new (Siehe Seite 94.)	–lay.response (Siehe Seite 98.)
–process (nur XSLT) (Siehe Seite 94.)	–lop (Siehe Seite 98.)
–scriptnames (Siehe Seite 94.)	–max (Siehe Seite 98.)
–view (Siehe Seite 94.)	–modid (Siehe Seite 98.)
	–query (Siehe Seite 99.)
	–recid (Siehe Seite 100.)
	–relatedsets.filter (Siehe Seite 100.)
	–relatedsets.max (Siehe Seite 101.)
	–script (Siehe Seite 101.)
	–script.param (Siehe Seite 101.)
	–script.prefind (Siehe Seite 102.)
	–script.prefind.param (Siehe Seite 102.)
	–script.presort (Siehe Seite 102.)
	–script.presort.param (Siehe Seite 102.)
	–skip (Siehe Seite 103.)
	–sortfield.[1-9] (Siehe Seite 103.)
	–sortorder.[1-9] (Siehe Seite 103.)
	–stylehref (Siehe Seite 104.)
	–styletype (Siehe Seite 105.)
	–token.[string] (nur XSLT) (Siehe Seite 105.)

Wichtig Der Parameter –lay für die Angabe eines Datenbanklayouts ist bei allen Query-Befehlen erforderlich mit Ausnahme von –dbnames, –layoutnames, –scriptnames und –process (nur XSLT-Abfragen).

Richtlinien für die Verwendung von Query-Befehlen und -Parametern

Beachten Sie folgende Richtlinien, wenn Sie Query-Befehle und -Parameter in einem Query-String verwenden:

- Ein Query-String darf nur einen einzigen Query-Befehl enthalten. Ein Query-String kann z. B. –new enthalten, um einen neuen Datensatz hinzuzufügen, aber derselbe Query-String kann nicht gleichzeitig –new und –edit enthalten.
- Die meisten Query-Befehle verlangen mehrere entsprechende Query-Parameter im Query-String. Beispielsweise verlangen alle Query-Befehle außer –dbnames und –process den Parameter –db, der die abzufragende Datenbank angibt. (Siehe dazu die Tabellen der erforderlichen Parameter in „Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings“ auf Seite 38.)
- Geben Sie für Query-Parameter und Feldnamen den jeweils gewünschten Wert an, z. B. –db=employees. Geben Sie für Query-Befehle kein =-Zeichen oder einen Wert nach dem Befehlsnamen wie –findall an.
- Query-Befehls- und -Parameternamen müssen in Kleinbuchstaben, z. B. –delete oder –lay, angegeben werden.
- Datenbanknamen, Layoutnamen und Feldnamen, die in Query-Strings verwendet werden, unterscheiden keine Groß- und Kleinschreibung, z. B. gibt –lay=meinlayout den Layoutnamen MeinLayout an.

  Hinweis Feld- und Datenbanknamen, die in XSLT-Anweisungen außerhalb von Query-Strings verwendet werden, unterscheiden Groß-/Kleinschreibung und müssen exakt mit den tatsächlich in der Datenbank verwendeten Namen übereinstimmen. In dieser Anweisung:

  ```
  <xsl:value-of select="fmrs:field[@name='LastName']"/>
  ```

  Der Feldverweis LastName in diesem Beispiel muss exakt mit dem Namen des Felds LastName in der Datenbank übereinstimmen.

- Feldnamen können Punkte enthalten, es gelten jedoch folgende Ausnahmen:
  - Dem Punkt darf keine Zahl folgen. Beispielsweise ist meinfeld.9 ein ungültiger Dateiname.
  - Dem Punkt darf nicht die Textzeichenfolge op (die beiden Buchstaben o und p) folgen. Beispielsweise ist meinfeld.op ein ungültiger Feldname.
  - Dem Punkt darf nicht die Textzeichenfolge global (das Wort „global“) folgen. Beispielsweise ist meinfeld.global ein ungültiger Feldname.

  Auf Feldnamen, die eine dieser Ausnahmen enthalten, kann nicht über XML oder XSLT mithilfe einer HTTP-Query zugegriffen werden. Diese Konstrukte sind für Datensatz-IDs, wie nachfolgend im Abschnitt „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ beschrieben, reserviert.
- Für den Befehl –find wird die Groß-/Kleinschreibung eines Feldwerts nicht berücksichtigt. Sie können beispielsweise Field1=Blue oder Field1=blue verwenden. Für die Befehle –new und –edit wird die verwendete Schreibung für einen Feldwert beibehalten und exakt so in der Datenbank gespeichert wie im Query-String angegeben. Beispiel: LastName=Doe.

Erläuterung der Referenz von FileMaker-Query-Strings

Diese Version enthält eine FileMaker-Datenbank mit dem Namen Query Strings Reference.fp7, die kurze Beschreibungen und Beispiele für jeden der FileMaker-Query-Befehle und -Query-Parameter bietet. Diese finden Sie im folgenden Verzeichnis auf jedem Rechner Ihres FileMaker Server-Einsatzes (Master- oder Arbeitscomputer):

Mac:

/Library/FileMaker Server/Examples/XSLT

Windows:

<Laufwerk>:\Programme\FileMaker\FileMaker Server\Examples\XSLT

Hierbei gilt: <Laufwerk> das Hauptlaufwerk ist, von dem das System startet.

Erläuterung der Syntax für einen voll qualifizierten Feldnamen

Ein vollständig qualifizierter Feldname identifiziert eine exakte Instanz eines Felds. Da Felder mit gleichen Namen auf unterschiedlichen Tabellen basieren können, müssen Sie vollständig qualifizierte Namen verwenden, um Fehler zu vermeiden.

Syntax für einen vollständig qualifizierten Feldnamen:

tabellen-name::feld-name(wiederholung-nummer).datensatz-id

Dabei gilt:

- tabellen-name ist der Name der Tabelle, die das Feld enthält. Der Tabellenname ist nur erforderlich, wenn sich das Feld nicht in der zugrunde liegenden Tabelle des im Query-String angegebenen Layouts befindet.
- feld-name(wiederholung-nummer) ist der spezifische Wert in einem Wiederholfeld und ist nur für Wiederholfelder erforderlich. Die Wiederholungsnummer wird ab der Ziffer 1 gezählt. Beispielsweise gibt feld-name(2) den zweiten Wert in einem Wiederholfeld an. Wenn Sie keine Wiederholungsnummer für ein Wiederholfeld angeben, wird der erste Wert des Wiederholfelds verwendet. Die Wiederholungsnummer ist für die Query-Befehle –new und –edit erforderlich, wenn sie Wiederholfelder verwenden, jedoch nicht für den Befehl –find.
- datensatz-id gibt die Datensatz-ID an. Diese ist nur erforderlich, wenn Sie einen Query-String verwenden, um Datensätze in einem Ausschnitt hinzuzufügen oder zu bearbeiten. Weitere Informationen finden Sie in den Abschnitten „Hinzufügen von Datensätzen in einem Ausschnitt“ und “Bearbeiten von Datensätzen in einem Ausschnitt.” Die record-id ist für die Query-Befehle –new und –edit erforderlich, wenn sie Ausschnittfelder verwenden, jedoch nicht für den Befehl –find.

Hinweis Damit sie verfügbar sind, müssen Felder auf das Layout gesetzt werden, das Sie im Query-String angeben.

Verwenden von Query-Befehlen für Ausschnittfelder

Die folgenden Abschnitte beschreiben, wie Query-Befehle für Ausschnittfelder funktionieren.

Hinzufügen von Datensätzen in einem Ausschnitt

Um einem Ausschnitt gleichzeitig mit dem Hinzufügen eines Ausgangsdatensatzes einen neuen Datensatz hinzuzufügen, verwenden Sie den Query-Befehl –new und führen den folgenden Query-String für die Abfrage aus:

- Verwenden Sie den vollständig qualifizierten Feldnamen für das Bezugsfeld im Ausschnitt.
- Geben Sie 0 als Datensatz-ID nach dem Namen des Bezugsfelds im Ausschnitt an.

- Geben Sie mindestens eines der Felder für den Ausgangsdatensatz an, bevor Sie das Bezugsfeld im Ausschnitt angeben.
- Geben Sie die Daten für das Abgleichsfeld (Schlüsselfeld) im Ausgangsdatensatz an.

Beispiel: Der folgende URL fügt einen neuen Employee-Ausgangsdatensatz für John Doe und gleichzeitig einen neuen Bezugsdatensatz für Jane im Ausschnitt hinzu. Der Name der Bezugstabelle ist „Dependents“ und der Name des Bezugsfelds im Ausschnitt lautet „Names“. Das Abgleichsfeld ID speichert eine Angestelltenkennung.

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&FirstName=John&LastName=Doe&ID=9756&Dependents::Names.0=Jane&–new

Hinweis Sie können pro Abfrage nur einen Bezugsdatensatz hinzufügen.

Bearbeiten von Datensätzen in einem Ausschnitt

Zur Bearbeitung eines oder mehrerer Datensätze in einem Ausschnitt verwenden Sie den Befehl –edit und eine Datensatz-ID, um den Ausgangsdatensatz anzugeben, der die zu bearbeitenden Ausschnitt-Datensätze enthält. Geben Sie den entsprechenden zu bearbeitenden Ausschnitt-Datensatz anhand seiner Datensatz-ID in einem vollständig qualifizierten Feldnamen an. Sie können eine Datensatz-ID dem Datensatz-ID-Attribut des <record>-Elements im <relatedset>-Element in den XML-Daten entnehmen. Weitere Informationen finden Sie unter „Verwenden der fmresultset-Grammatik“ auf Seite 32.

Beispiel: Der folgende URL bearbeitet einen Datensatz in einem Ausschnitt, wobei der Ausgangsdatensatz die Datensatz-ID 1001 hat. Der Name der Bezugstabelle ist „Dependents“, der Name des Bezugsfelds im Ausschnitt lautet „Names“ und die 2 in „Names.2“ ist die Datensatz-ID des Ausschnitt-Datensatzes.

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–recid=1001&Dependents::Names.2=Kevin&–edit

Das folgende Beispiel zeigt, wie Sie eine Abfrage verwenden können, um mehrere Ausschnitt-Datensätze über den Ausgangsdatensatz zu bearbeiten:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–recid=1001&Dependents::Names.2=Kevin&Dependents::Names.5=Susan&–edit

Sie können auch den Befehl –edit verwenden und 0 als ID des Ausschnitt-Datensatzes angeben, um dem Ausschnitt einen neuen Bezugsdatensatz für einen bestehenden Ausgangsdatensatz hinzuzufügen. Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–recid=1001&Dependents::Names.0=Timothy&–edit

Löschen von Ausschnittdatensätzen

Um Ausschnittdatensätze zu löschen, verwenden Sie den Parameter –delete.related mit dem Befehl –edit anstelle des Befehls –delete.

Zum Beispiel löscht der folgende URL den Datensatz 1001 aus der Tabelle employees:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–recid=1001&–delete

Aber der folgende URL löscht einen Ausschnittdatensatz mit einer Datensatz-ID 3 aus der Bezugstabelle Dependents mit der Ausgangsdatensatz-ID 1001.

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–recid=1001&–delete.related=Dependents.3&–edit

Weitere Informationen finden Sie unter „Query-Parameter –delete.related (Löschen von Ausschnittdatensätzen)“ auf Seite 95.

Abfragen von Ausschnittfeldern

In einer Lösung mit vielen Datensätzen kann das Abfragen und Sortieren von Datensätzen einige Zeit in Anspruch nehmen. Um die anzuzeigende Anzahl von Datensätzen und Zeilen in einer Bezugsmenge einzuschränken, verwenden Sie die Parameter –relatedsets.filter und –relatedsets.max bei Suchabfragen. Weitere Informationen finden Sie unter „Query-Parameter –relatedsets.filter (Ausschnittdatensätze filtern)“ auf Seite 100 und „Query-Parameter –relatedsets.max (Ausschnittdatensätze beschränken)“ auf Seite 101.

Erläuterung der Syntax für das Angeben eines Variablenfelds

Syntax für das Angeben eines Variablenfelds:

tabellen-name::feld-name(wiederholung-nummer).global

Dabei gilt: global gibt ein Feld an, das global gespeichert wird. Informationen über tabellen-name und feld-name(wiederholung-nummer) finden Sie unter „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 89. Informationen über Variablenfelder finden Sie in der FileMaker Pro Hilfe.

Sie müssen die .global-Syntax verwenden, um ein Variablenfeld in einem Query-String zu identifizieren. Die Web Publishing Engine setzt die Parameterwerte für Variablenfelder, bevor sie den Query-Befehl ausführt oder andere Parameterwerte im Query-String setzt. Für direkte XML-Abfragen sowie Abfragen, die über XSLT-Stylesheets erfolgen und keine Sitzungen verwenden, laufen die globalen Werte sofort nach Ausführung der Abfrage ab. Für Abfragen, die über XSLT-Stylesheets erfolgen und Sitzungen verwenden, bleiben die globalen Werte für die Dauer der Sitzung erhalten, die im Stylesheet definiert ist, oder bis sie erneut durch eine andere Abfrage geändert werden.

Wenn Sie ein Variablenfeld in einem Query-String nicht mit der .global-Syntax identifizieren, berechnet die Web Publishing Engine das Variablenfeld zusammen mit dem restlichen Query-String, ohne zuvor den Wert für das Variablenfeld zu setzen.

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&Country.global=USA&–edit

Hinweis Wenn Sie den Wert eines Variablenfelds mithilfe des Befehls –edit setzen, erübrigt sich der Parameter –recid, sofern Sie mit dieser Abfrage nur den Wert des Variablenfelds setzen.

Wichtig Wenn Sie ein Variablenfeld in einem XSLT-Stylesheet benutzen, müssen Sie mithilfe von Admin Console die Option XSLT-Datenbanksitzungen für die Web Publishing Engine aktivieren. Andernfalls werden Variablenfeldwerte zwischen Abfragen nicht beibehalten. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

Query-Befehlsreferenz

Dieser Abschnitt enthält Informationen über die Query-Befehle, die für XML- und XSLT-Abfragen zur Verfügung stehen.

Hinweis Nur in XSLT-Abfragen erfordern alle folgenden Query-Befehle den Query-Parameter –grammar.

Query-Befehl –dbnames (Datenbanknamen)

Ruft die Namen aller Datenbanken ab, die FileMaker Server bereitstellt und die für Custom Web Publishing mit XML oder XSLT aktiviert sind.

Erforderliche Query-Parameter: (keine)

Beispiel:

So rufen Sie die Datenbanknamen ab:

http://192.168.123.101/fmi/xml/fmresultset.xml?–dbnames

Query-Befehl –delete (Datensatz löschen)

Löscht den vom –recid-Parameter angegebenen Datensatz.

Erforderliche Query-Parameter: –db, –lay, –recid

Optionaler Query-Parameter: –script

Beispiel:

So löschen Sie einen Datensatz:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–recid=4&–delete

Query-Befehl –dup (Datensatz duplizieren)

Dupliziert den durch –recid angegebenen Datensatz.

Erforderliche Query-Parameter: –db, –lay, –recid

Optionaler Query-Parameter: –script

Beispiel:

So duplizieren Sie den angegebenen Datensatz:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–recid=14&–dup

Query-Befehl –edit (Datensatz bearbeiten)

Aktualisiert den vom Parameter –recid angegebenen Datensatz und füllt die Felder mit dem Inhalt etwaiger Feldname-Wert-Paare. Der –recid-Parameter gibt an, welcher Datensatz bearbeitet werden sollte.

Erforderliche Query-Parameter: –db, –lay, –recid, ein oder mehrere Feldname(n)

Optionaler Query-Parameter: –modid, –script

Hinweis Informationen über das Bearbeiten von Datensätzen in einem Ausschnitt finden Sie unter „Bearbeiten von Datensätzen in einem Ausschnitt“ auf Seite 90.

Beispiel:

So bearbeiten Sie einen Datensatz:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–recid=13&Country=USA&–edit

Query-Befehle –find, –findall oder –findany (Datensätze suchen)

Sendet eine Suchabfrage mit definierten Kriterien.

Erforderliche Query-Parameter: –db, –lay

Optionale Query-Parameter: –recid, –lop, –op, –max, –skip, –sortorder, –sortfield, –script, –script.prefind, –script.presort, Feldname

Beispiele:

So suchen Sie einen Datensatz nach Feldname:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&Country=USA&–find

Hinweis Die mehrfache Angabe eines Feldnamens in einer einzelnen Abfrage wird nicht unterstützt. FileMaker Server geht alle Werte durch, verwendet aber nur den zuletzt geparsten Wert.

So suchen Sie einen Datensatz nach Datensatz-ID:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–recid=427&–find

Um alle Datensätze in der Datenbank zu finden, verwenden Sie –findall:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–findall

Um einen zufällig ermittelten Datensatz zu finden, verwenden Sie –findany:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–findany

Query-Befehl –findquery (Compound-Suche)

Sendet eine Suchabfrage mit mehreren Suchdatensätzen und Datensatzausschluss-Abfragen.

Erforderliche Query-Parameter: –db, –lay, –query

Optionale Query-Parameter: –max, –skip, –sortorder, –sortfield, –script, –script.prefind, –script.presort

Beispiel:

Sucht Datensätze mit Hunden oder Katzen, die nicht „Fluffy“ heißen.

http://host/fmi/xml/fmresultset.xml?–db=vetclinic&–lay=animals&–query=(q1);(q2);!(q3)
&–q1=typeofanimal&–q1.value=Cat&–q2=typeofanimal&–q2.value=Dog&–q3=name&–q3.value=Fluffy&–findquery

Verwenden des Befehls -findquery für Compound-Suchen

Eine Anweisung –findquery besteht aus vier Teilen in der folgenden Reihenfolge:

- Der Parameter –query.
- Die Query-Abfragedeklarationen, bestehend aus den Query-Bezeichnerdeklarationen und Abfragevorgängen.
- Das Suchfeld und Wertedefinitionen für jeden Query-Bezeichner.
- Der Befehl –findquery am Ende der gesamten Anweisung.

Weitere Informationen zur Verwendung des Parameters –query finden Sie unter „Query-Parameter –query (Compound-Suchabfrage)“ auf Seite 99.

Query-Befehl –layoutnames (Layoutnamen)

Ruft die Namen aller verfügbaren Layouts für eine angegebene Datenbank ab, die FileMaker Server bereitstellt und die für Custom Web Publishing mit XML oder XSLT aktiviert sind.

Erforderliche Query-Parameter: –db

Beispiel:

So rufen Sie die Namen verfügbarer Layouts ab:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–layoutnames

Query-Befehl –new (Neuer Datensatz)

Erzeugt einen neuen Datensatz und fügt in diesen den Inhalt jeglicher Feldnamen/Wert-Paare ein.

Erforderliche Query-Parameter: –db, –lay

Optionaler Query-Parameter: ein oder mehrere Feldname(n), –script

Hinweis Informationen über das Einfügen neuer Daten für einen Ausschnitt finden Sie unter „Hinzufügen von Datensätzen in einem Ausschnitt“ auf Seite 89.

Beispiel:

So fügen Sie einen neuen Datensatz hinzu:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&Country=Australia&–new

–process (XSLT-Stylesheets verarbeiten)

Verarbeitet ein XSLT-Stylesheet, ohne Daten von der Datenbank anzufordern. Dieser Query-Befehl kann nur mit XSLT-Stylesheets verwendet werden.

Erforderlicher Query-Parameter: –grammar Sie müssen die fmresultset- oder FMPXMLRESULT-Grammatik verwenden.

Beispiel:

http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xsl?–grammar=fmresultset&–process

Weitere Informationen finden Sie unter „Verarbeitung von XSLT-Abfragen, die nicht FileMaker Server abfragen“ auf Seite 58.

Query-Befehl –scriptnames (Scriptnamen)

Ruft die Namen aller verfügbaren Scripts für eine angegebene Datenbank ab, die FileMaker Server bereitstellt und die für Custom Web Publishing mit XML oder XSLT aktiviert sind.

Erforderliche Query-Parameter: –db

Beispiel:

So rufen Sie die Namen aller Scripts ab:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–scriptnames

Query-Befehl –view (Layoutinformationen anzeigen)

Wenn die FMPXMLLAYOUT-Grammatik angegeben ist: Ruft Layoutinformationen aus einer Datenbank ab und zeigt sie in der FMPXMLLAYOUT-Grammatik an. Wenn eine Datengrammatik (fmresultset oder FMPXMLRESULT) angegeben ist: Ruft den Metadaten-Abschnitt des XML-Dokuments und einen leeren „recordset“ ab.

Erforderliche Query-Parameter: –db, –lay

Optionaler Query-Parameter: –script

Beispiele:

So rufen Sie Layoutinformationen ab:

http://192.168.123.101/fmi/xml/FMPXMLLAYOUT.xml?–db=employees&–lay=departments&–view

So rufen Sie Metadaten-Informationen ab:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–view

Query-Parameterreferenz

Dieser Abschnitt enthält Informationen über die Query-Parameter, die für XML- und XSLT-Abfragen zur Verfügung stehen. Informationen über Parameter, die nur für XSLT-Abfragen verfügbar sind, finden Sie unter „Verwenden von Query-Strings in FileMaker XSLT-Stylesheets“ auf Seite 54.

Query-Parameter –db (Datenbankname)

Gibt die Datenbank an, auf die der Query-Befehl angewendet wird.

Wert: Name der Datenbank ohne etwaige Dateinamenerweiterung.

Hinweis Geben Sie mit dem Namen der Datenbank für den Parameter –db keine Dateinamenerweiterung an. Der tatsächliche Datenbank-Dateiname kann optional eine Erweiterung aufweisen, aber Erweiterungen sind als Wert für den Parameter –db nicht erlaubt.

Erforderlich für: Alle Query-Befehle mit Ausnahme von –dbnames und –process

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–findall

Query-Parameter –delete.related (Löschen von Ausschnittdatensätzen)

Löscht einen Datensatz aus einem Ausschnittfeld.

Optional für: Query-Befehl –edit

Erfordert: Ein Bezugstabellenname und eine Datensatz-ID

Beispiel:

Das folgende Beispiel löscht einen Ausschnittdatensatz mit einer Datensatz-ID 20 aus der Bezugstabelle jobtable mit der Ausgangsdatensatz-ID 7.

http://host/fmi/xml/fmresultset.xml?–db=career&–lay=applications&–recid=7&–delete.related=jobtable.20&–edit

Query-Parameter –encoding (XSLT-Abfrage-Kodierung)

Gibt die Textkodierung für eine XSLT-Abfrage an. Dieser Query-Befehl kann nur für Custom Web Publishing mit XSLT-Abfragen verwendet werden.

Wert: US-ASCII, ISO-8859-1, ISO-8859-15, ISO-2022-JP, Shift_JIS, or UTF-8

Optional für: Alle Query-Befehle in einer XSLT-Abfrage

Beispiel:

http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xsl?–db=employees&–lay=departments
&–grammar=fmresultset&–encoding=Shift_JIS&–findall

Weitere Informationen finden Sie unter „Festlegen von Textkodierung für Abfragen“ auf Seite 57.

Query-Parameter –field (Medienfeldname)

Gibt den Namen eines Medienfelds zurück.

Erforderlich für: Abfrage von Daten in einem Medienfeld.

Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XML-Lösungen“ auf Seite 28, und „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen“ auf Seite 53.

Query-Parameter „fieldname“ (Nicht-Medienfeldname)

Feldnamen dienen als Kontrollkriterien für den Query-Befehl –find oder zum Ändern des Inhalts eines Datensatzes. Wenn ein Query-Befehl oder -Parameter einen Wert für ein Nicht-Medienfeld verlangt, verwenden Sie den Feldnamen ohne den Bindestrich (–) als den Namensteil des Name-Wert-Paars.

Name: Name des Felds in der FileMaker-Datenbank. Wenn das Feld sich nicht in der zugrunde liegenden Tabelle des Layouts befindet, das im Query-String angegeben ist, muss der Feldname vollständig qualifiziert sein. Feldnamen können Punkte enthalten, es gelten jedoch folgende Ausnahmen:

- Dem Punkt darf keine Zahl folgen. Beispielsweise ist meinfeld.9 ein ungültiger Dateiname.
- Dem Punkt darf nicht die Textzeichenfolge op (die beiden Buchstaben o und p) folgen. Beispielsweise ist meinfeld.op ein ungültiger Feldname.
- Dem Punkt darf nicht die Textzeichenfolge global (das Wort „global“) folgen. Beispielsweise ist meinfeld.global ein ungültiger Feldname.

Auf Feldnamen, die eine dieser Ausnahmen enthalten, kann nicht über XML oder XSLT mithilfe einer HTTP-Query zugegriffen werden. Diese Konstrukte sind für Datensatz-IDs wie in Abschnitt „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 89 beschrieben reserviert.

Wert: Geben Sie für die Query-Befehle –new und –edit den Wert an, den Sie im aktuellen Datensatz in dem Feld speichern wollen. Geben Sie für den Query-Befehl –find den Wert an, den Sie im Feld suchen wollen. Wenn Sie den Wert für ein Datums-, Zeit- oder Zeitstempelfeld angeben, müssen Sie den Wert gemäß „fm“-Format für den Feldtyp angeben. Die „fm“-Formate sind MM/tt/jjjj für Datum, HH:mm:ss für Zeit und MM/tt/jjjj HH:mm:ss für Zeitstempel.

Erforderlich für: –edit (Query-Befehl)

Optional für: Query-Befehle –new und –find

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–op=eq&FirstName=Sam&–max=1&–find

Hinweis Die mehrfache Angabe eines Feldnamens in einer einzelnen Abfrage wird nicht unterstützt. FileMaker Server geht alle Werte durch, verwendet aber nur den zuletzt geparsten Wert.

Query-Parameter „fieldname.Op“ (Vergleichsoperator)

Gibt den Vergleichsoperator an, der auf den Feldnamen, der dem Operator vorangeht, angewendet werden soll. Vergleichsoperatoren werden mit dem Query-Befehl –find verwendet.

Wert: Der zu verwendende Operator. Der Standard-Operator lautet „begins with“. Gültige Operatoren sind:

Schlüsselwort	Äquivalenter Operator in FileMaker Pro
eq	= Wort
cn	*Wort*
bw	Wort*
ew	*Wort
gt	> Wort
gte	>= Wort
lt	< Wort
lte	<= Wort
neq	ausschließen, Wort

Optional für: Query-Befehl –find

Erfordert: Einen Feldnamen und einen Wert

Syntax für das Angeben eines Vergleichsoperators:

tabellen-name::feld-name=wert&tabelle-name::feld-name.op=op-symbol

Dabei gilt:

- tabelle-name ist die Tabelle, die das Feld enthält, und ist nur erforderlich, wenn das Feld nicht in der Quelltabelle des im Query-String angegebenen Layouts ist.
- op-symbol ist eines der Schlüsselwörter in der vorherigen Tabelle, wie z. B. cn.

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&name=Tim&name.op=cn&–find

Sie können jeden FileMaker Pro-Suchoperator verwenden, indem Sie das Schlüsselwort bw angeben. Um z. B. den Wertebereich über den Bereichsoperator (...) zu ändern, geben Sie das Schlüsselwort bw an und setzen dann die Zeichen „...“ vor das Suchkriterium.

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&IDnum=915...925&IDnum.op=bw&–find

Weitere Informationen über Operatoren für die Suche nach Text finden Sie in der FileMaker Pro Hilfe.

Query-Parameter –grammar (Grammatik für XSLT-Stylesheets)

Gibt die Grammatik für ein XSLT-Stylesheet an. Dieser Query-Befehl kann nur für Custom Web Publishing mit XSLT-Abfragen verwendet werden.

Wert: fmresultset, FMPXMLRESULT oder FMPXMLLAYOUT

Erforderlich für: Alle XSLT-Abfragen

Beispiel:

http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xsl?–grammar=fmresultset&–db=mydatabase&–lay=mylayout&–findall

Weitere Informationen finden Sie unter „Angeben einer XML-Grammatik für ein FileMaker XSLT-Stylesheet“ auf Seite 55.

Query-Parameter –lay (Layout)

Gibt das Datenbanklayout an, das Sie verwenden wollen.

Wert: Name des Layouts.

Erforderlich für: Alle Query-Befehle mit Ausnahme von –dbnames, –layoutnames, –scriptnames und –process (nur XSLT-Abfragen)

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–view

Query-Parameter –lay.response (Layout wechseln für Antwort)

Gibt an, dass FileMaker Server beim Verarbeiten einer Abfrage das Layout verwenden soll, das der Parameter –lay angibt, und bei der Verarbeitung der XML-Antwort in das Layout wechseln soll, das der Parameter –lay.response angibt.

Wenn Sie den Parameter –lay.response nicht angeben, verwendet FileMaker Server das vom Parameter –lay angegebene Layout für die Verarbeitung der Abfrage und der Antwort.

Sie können den Parameter –lay.response für XML-Abfragen oder in XSLT-Stylesheet-Abfragen verwenden.

Wert: Name des Layouts.

Optional für: Alle Query-Befehle mit Ausnahme von –dbnames, –layoutnames, –scriptnames und –process (nur XSLT-Abfragen)

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=Budget&Salary=100000&Salary.op=gt&–find&–lay.response=VorstListe

Query-Parameter –lop (Logischer Operator)

Gibt an, wie die Suchkriterien im Query-Befehl –find als and- oder or-Suche kombiniert werden.

Wert: and oder or (müssen in Kleinbuchstaben angegeben werden). Wenn der Query-Parameter –lop nicht angegeben wird, verwendet der Query-Befehl –find den Wert and.

Optional für: Query-Befehl –find

Hinweis Nicht unterstützt von Query-Befehl -findquery.

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&Last+Name=Smith&Birthdate=2/5/1972&–lop=and&–find

Query-Parameter –max (Maximum Datensätze)

Gibt die maximale Anzahl an Datensätzen an, die zurückgegeben werden soll.

Wert: Eine Zahl oder – für die Rückgabe aller Datensätze – der Wert all. Der Wert all muss in Kleinbuchstaben angegeben werden. Wenn –max nicht angegeben wird, werden alle Datensätze zurückgegeben.

Optional für: Query-Befehle –find, –findall und -findquery

Beispiele:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–max=10&–findall

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–max=all&–findall

Query-Parameter –modid (Änderungs-ID)

Die Änderungs-ID ist ein inkrementeller Zähler, der die aktuelle Version eines Datensatzes angibt. Indem Sie mit dem Query-Befehl –edit eine Änderungs-ID angeben, können Sie sicherstellen, dass Sie die aktuelle Version eines Datensatzes bearbeiten. Wenn der angegebene Wert der Änderungs-ID nicht dem aktuellen Wert der Änderungs-ID in der Datenbank entspricht, wird der Query-Befehl –edit nicht zugelassen und ein Fehlercode wird zurückgegeben.

Wert: Eine Änderungs-ID, die die aktuelle Version eines Datensatzes in einer FileMaker-Datenbank eindeutig identifiziert.

Optional bei: –edit (Query-Befehl)

Erfordert: –recid-Parameter

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–recid=22&–modid=6&last_name=Jones&–edit

Query-Parameter –query (Compound-Suchabfrage)

Gibt die Query-Namen und Suchkriterien für eine Compound-Suchabfrage an. Weitere Informationen finden Sie unter „Query-Befehl –findquery (Compound-Suche)“ auf Seite 93.

Wert: Ein Query-Ausdruck.

Erforderlich für: Query-Befehl –findquery

Die Syntax für eine Compound-Suchabfrage ist:

–query=<request-declarations><request-definitions>&–findquery

Hierbei gilt:

<request-declarations> sind zwei oder mehr Abfragedeklarationen.

- Jede Abfragedeklaration setzt sich aus einem oder mehreren Query-Bezeichnern zusammen, die durch Kommata getrennt und in Klammern eingeschlossen sind. Ein Query-Bezeichner ist der Buchstabe „q“ gefolgt von einer Zahl. Beispiel: q1
- Eingeschlossen in Klammern fungieren mehrere Queries als logische UND-Suchen, die die Ergebnismenge einengen. Zum Beispiel gibt (q1, q2) Datensätze zurück, die q1 und q2 entsprechen.
- Abfragen werden durch Semikolon getrennt. Mehrere Abfragen fungieren als logische ODER-Suchen, die die Ergebnismenge erweitern. Zum Beispiel gibt (q1);(q2) Datensätze zurück, die q1 oder q2 entsprechen.
- Wie bei FileMaker Pro kann jede Abfrage eine Such- oder Ausschlussabfrage sein. Eine Suchabfrage fügt die entsprechenden Datensätze der Ergebnismenge hinzu. Eine Ausschlussabfrage entfernt die entsprechenden Datensätze aus der Ergebnismenge. Standard ist eine Suchabfrage. Setzen Sie für eine Ausschlussabfrage ein Ausrufezeichen (!) vor die öffnende Klammer.

  Beispiel: (q1);!(q2)

  In diesem Beispiel ist q1 eine Suchabfrage. q2 ist eine Ausschlussabfrage, da ihr ein Ausrufezeichen vorangestellt ist.
- Abfragen werden durch Semikolon getrennt. Mehrere Suchabfragen fungieren als logische ODER-Suchen, die die Ergebnismenge erweitern. Zum Beispiel gibt (q1);(q2) Datensätze zurück, die q1 oder q2 entsprechen. Ausschlussabfragen fungieren nicht als logische ODER-Suchen, da Ausschlussabfragen Datensätze aus der Ergebnismenge entfernen.
- Abfragen werden in der angegebenen Reihenfolge ausgeführt. Die Ergebnismenge umfasst die Ergebnisse aus der gesamten zusammengesetzten Suchabfrage.

<request-definitions> ist eine Abfragedefinition für jede Abfragedeklaration. Jede Abfragedefinition besteht aus Suchfeld und Wertdefinition. Ein Minuszeichen (–) beginnt die Abfragedefinition.

Syntax:

–<*query-id*>=<*fieldname*>&–<*query-id*>.value=<*value*>

Beispiel:

–q1=typeofanimal&–q1.value=Cat

–q2=name&–q2.value=Fluffy

Beispiel:

Sucht Datensätze mit grauen Katzen, die nicht „Fluffy“ heißen.

http://host/fmi/xml/fmresultset.xml?–db=petclinic&–lay=Patients&–query=(q1, q2);!(q3)
&–q1=typeofanimal&–q1.value=Cat&–q2=color&–q2.value=Gray&–q3=name&–q3.value=Fluffy&–findquery

Query-Parameter –recid (Datensatz-ID)

Gibt den Datensatz an, der verarbeitet werden soll. Wird hauptsächlich mit den Query-Befehlen –edit und –delete verwendet. Vom Befehl -view verwendet, um Bezugswertlistendaten in der Grammatik FMPXMLLAYOUT abzurufen.

Wert: Eine Datensatz-ID, die einen Datensatz in einer FileMaker-Datenbank eindeutig angibt.

Erforderlich für: Query-Befehle –edit, –delete und –dup

Optional bei Befehlen: –find und –view

Beispiel 1:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–recid=22&–delete

Beispiel 2:

http://localhost/fmi/xml/FMPXMLLAYOUT.xml?-db=test&-lay=empty&-view&-recid=9

Query-Parameter –relatedsets.filter (Ausschnittdatensätze filtern)

Gibt an, ob die zurückgegebenen Zeilen für Queries, die Ausschnittfelder verwenden, gefiltert werden.

Wert: layout oder none

- Der Standardwert ist none, wenn dieser Parameter nicht angegeben ist.
- Wenn die Query layout angibt, werden die im FileMaker Pro-Dialogfeld „Ausschnitteinstellungen“ angegebenen Einstellungen respektiert. Die Datensätze werden basierend auf der im Dialogfeld „Ausschnitteinstellungen“ definierten Sortierung sortiert, wobei die Datensatzmenge gefiltert ist, um mit der angegebenen ersten Zeile zu beginnen.
- Wenn die Einstellung „Vertikaler Rollbalken“ im Dialogfeld „Ausschnitteinstellungen“ aktiviert ist, können Sie die Option –relatedsets.max verwenden, um die maximale Anzahl an Zeilen anzugeben, die in der Antwort auf die Query zurückgegeben werden.
- Wenn „Vertikaler Rollbalken“ deaktiviert ist, bestimmt die Einstellung „Anzahl der Zeilen“ im FileMaker Pro-Dialogfeld „Ausschnitteinstellungen“ die Anzahl an anzuzeigenden Zeilen.
- Wenn –relatedsets.filter auf none eingestellt ist, gibt die Web Publishing Engine alle Zeilen im Ausschnitt zurück und die Ausschnittdatensätze werden nicht vorsortiert.

Optional bei: –find, –edit, –new, –dup und –findquery.

Beispiele:

http://localhost/fmi/xml/fmresultset.xml?–db=FMPHP_Sample&–lay=English&–relatedsets.filter=none&–findany

http://localhost/fmi/xml/fmresultset.xml?–db=FMPHP_Sample&–lay=English&relatedsets.filter=layout
&–relatedsets.max=all&–findany

http://localhost/fmi/xml/fmresultset.xml?–db=FMPHP_Sample&–lay=English&–relatedsets.filter=layout
&–relatedsets.max=10&–findany

Query-Parameter –relatedsets.max (Ausschnittdatensätze beschränken)

Gibt die maximale Anzahl an Zeilen an, die in den Ergebnissen für diese Query zurückgegeben werden.

Wert: eine Ganzzahl oder all.

- Wenn die Query eine Ganzzahl angibt, gibt die Web Publishing Engine die Anzahl an Zeilen nach der ersten Zeile zurück.
- Wenn die Query all angibt, gibt die Web Publishing Engine alle Bezugsdatensätze zurück.
- Wenn die Query den Parameter –relatedsets.max nicht angibt, wird die Anzahl der Zeilen durch den Wert bestimmt, der für den Parameter –relatedsets.filter angegeben ist. Weitere Informationen finden Sie unter „Query-Parameter –relatedsets.filter (Ausschnittdatensätze filtern)“ auf Seite 100.

Optional für: –find, –edit, –new, –dup und –findquery.

Beispiele:

http://localhost/fmi/xml/fmresultset.xml?–db=FMPHP_Sample&–lay=English&relatedsets.filter=layout&–relatedsets.max=all&–findany

http://localhost/fmi/xml/fmresultset.xml?–db=FMPHP_Sample&–lay=English&–relatedsets.filter=layout&–relatedsets.max=10&–findany

Query-Parameter –script (Script)

Gibt das FileMaker-Script an, das nach Ausführung des Query-Befehls und der Sortierung ablaufen soll. Weitere Informationen finden Sie unter „Erläuterung der Verarbeitung einer XML-Abfrage“ auf Seite 41.

Wert: Scriptname.

Optional für: Alle Query-Befehle mit Ausnahme von –dbnames, –layoutnames, –scriptnames und –process

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–script=myscript&–findall

Query-Parameter –script.param (Parameter an Script übergeben)

Übergibt einen Parameter an das durch –script angegebene FileMaker-Script.

Wert: Ein einzelner Textparameter.

- Um mehrere Parameter zu übergeben, können Sie einen String erstellen, der die Parameter begrenzt, und Ihr Script die einzelnen Parameter auslesen lassen. Übergeben Sie z. B. param1|param2|param3 als Liste mit dem Zeichen |, das so kodiert ist: param1%7Cparam2%7Cparam3
- Um den Textparameter als Wert zu behandeln, der kein Text ist, kann Ihr Script den Textwert konvertieren. Um z. B. den Textwert in eine Zahl zu konvertieren, könnte Ihr Script Folgendes beinhalten: LiesAlsZahl (Hole(ScriptParam))
- Wenn Ihre Query –script.param ohne –script enthält, wird –script.param ignoriert.
- Wenn Ihre Query mehr als einen –script.param enthält, verwendet die Web Publishing Engine den letzten geparsten Wert.

Optional für: –script

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–script=myscript&–script.param=Smith%7CChatterjee%7CSu&–findall

Query-Parameter –script.prefind (Script vor Suchen)

Gibt das FileMaker-Script an, das vor dem Suchen und Sortieren von Datensätzen (falls angegeben) während der Verarbeitung des Query-Befehls –find ausgeführt wird.

Wert: Scriptname.

Optional für: Alle Query-Befehle mit Ausnahme von –dbnames, –layoutnames, –scriptnames und –process

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–script.prefind=myscript&–findall

Query-Parameter –script.prefind.param (Parameter an Script vor Suchen übergeben)

Übergibt einen Parameter an das durch –script.prefind angegebene FileMaker-Script.

Wert: Ein einzelner Textparameter.

- Um mehrere Parameter zu übergeben, können Sie einen String erstellen, der die Parameter begrenzt, und Ihr Script die einzelnen Parameter auslesen lassen. Übergeben Sie z. B. param1|param2|param3 als Liste mit dem Zeichen |, das so kodiert ist: param1%7Cparam2%7Cparam3
- Um den Textparameter als Wert zu behandeln, der kein Text ist, kann Ihr Script den Textwert konvertieren. Um z. B. den Textwert in eine Zahl zu konvertieren, könnte Ihr Script Folgendes beinhalten: LiesAlsZahl (Hole(ScriptParam))
- Wenn Ihre Query –script.prefind.param ohne –script.prefind enthält, wird –script.prefind.param ignoriert.
- Wenn Ihre Query mehr als einen –script.prefind.param enthält, verwendet die Web Publishing Engine den letzten geparsten Wert.

Optional für: –script.prefind

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–script.prefind=myscript
&–script.prefind.param=payroll&–findall

Query-Parameter –script.presort (Script vor Sortieren)

Gibt das FileMaker-Script an, das nach dem Suchen von Datensätzen (falls angegeben) und vor dem Sortieren der Datensätze während der Verarbeitung des Query-Befehls –find ausgeführt wird.

Optional für: Alle Query-Befehle mit Ausnahme von –dbnames, –layoutnames, –scriptnames und –process

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–script.presort=myscript
&–sortfield.1=dept&–sortfield.2=rating&–findall

Query-Parameter –script.presort.param (Parameter an Script vor Sortieren übergeben)

Übergibt einen Parameter an das durch –script.presort angegebene FileMaker-Script.

Wert: Ein einzelner Textparameter.

- Um mehrere Parameter zu übergeben, können Sie einen String erstellen, der die Parameter begrenzt, und Ihr Script die einzelnen Parameter auslesen lassen. Übergeben Sie z. B. param1|param2|param3 als Liste mit dem Zeichen |, das so kodiert ist: param1%7Cparam2%7Cparam3
- Um den Textparameter als Wert zu behandeln, der kein Text ist, kann Ihr Script den Textwert konvertieren. Um z. B. den Textwert in eine Zahl zu konvertieren, könnte Ihr Script Folgendes beinhalten: LiesAlsZahl (Hole(ScriptParam))

- Wenn Ihre Query –script.presort.param ohne –script.presort enthält, wird –script.presort.param ignoriert.
- Wenn Ihre Query mehr als einen –script.presort.param enthält, verwendet die Web Publishing Engine den letzten geparsten Wert.

Optional für: –script.presort

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–script.presort=myscript&–script.presort.param=18%7C65&–sortfield.1=dept&–sortfield.2=rating&–findall

Query-Parameter –skip (Datensätze überspringen)

Gibt an, wie viele Datensätze in der Ergebnismenge übersprungen werden sollen.

Wert: Eine Zahl. Wenn der Wert größer als die Anzahl der Datensätze in der Ergebnismenge ist, wird kein Datensatz angezeigt. Der Standardwert beträgt 0.

Optional für: Query-Befehl –find

Im folgenden Beispiel werden die ersten 10 Datensätze in der Ergebnismenge ausgelassen und die Datensätze 11 bis 15 ausgegeben.

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–skip=10&–max=5&–findall

Query-Parameter –sortfield (Sortierfeld)

Gibt das Feld an, das für das Sortieren verwendet wird.

Wert: Feldname

Optional für: Query-Befehle –find oder –findall

Der Query-Parameter –sortfield kann mehrfach verwendet werden, um mehrere Feldsortierungen vorzunehmen. Syntax für die Rangfolge der Sortierfelder:

–sortfield.precedence-number=fully-qualified-field-name

Dabei gilt: rang-nummer im Query-Parameter –sortfield.rang-nummer ist eine Zahl (beginnend mit 1), die die Rangfolge für mehrere Sortierfelder angibt.

In folgendem Beispiel wird zuerst das Feld dept sortiert und dann das Feld rating. Beide Felder werden aufsteigend sortiert, da der Query-Parameter –sortorder nicht angegeben wurde.

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=performance&–sortfield.1=dept&–sortfield.2=rating&–findall

Query-Parameter –sortorder (Sortierfolge)

Gibt die Richtung einer Sortierung an.

Wert: Die Sortierfolge. Gültige Sortierfolgen (wobei <wertelisten-name> der Name einer Werteliste wie Spezial ist):

Schlüsselwort	Äquivalenter Operator in FileMaker Pro
ascend	Sortieren von a bis z, -10 bis 10
descend	Sortieren von z bis a, 10 bis -10
<wertelisten-name>	Sortieren nach der angegebenen Werteliste für das Feld im Layout

Optional für: Query-Befehle –find oder –findall

Erfordert: –sortfield (Query-Parameter)

Der Query-Parameter –sortorder kann mit dem Query-Parameter –sortfield verwendet werden, um die Sortierfolge mehrerer Sortierfelder anzugeben. Syntax für die Sortierfolge eines Sortierfelds:

–sortorder.precedence-number=sort-method

Dabei gilt:

- precedence-number im Query-Parameter –sortorder.precedence-number ist eine Zahl von 1 bis 9, die den Query-Parameter –sortfield angibt, auf den sich der Query-Parameter –sortorder bezieht.
- sortier-methode ist eines der Schlüsselwörter in obiger Tabelle für die Festlegung der Sortierfolge, z. B. „ascend“.

Im folgenden Beispiel ist die Sortierfolge für das Sortierfeld mit der höchsten Rangnummer (dept) „ascend“ und für das Sortierfeld mit der zweithöchsten Rangnummer (rating) „descend“. Die precedence-number 2 in –sortorder.2 gibt an, dass der Query-Parameter –sortorder.2=descend sich auf den Query-Parameter –sortfield.2=rating bezieht.

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=performance&–sortfield.1=dept
&–sortorder.1=ascend&–sortfield.2=rating&–sortorder.2=descend&–findall

Hinweis Ein Sortierfeld, für das der Query-Parameter –sortorder nicht angegeben ist, verwendet standardmäßig aufsteigende Sortierung.

Query-Parameter –stylehref (Stil-Href)

Generiert eine Verarbeitungsanweisung für XML-Stylesheets im Ausgabedokument. Dabei setzt er den Wert des href-Attributs (href=/mystylesheet.css or href=/stylesheets/mystylesheet.xsl), damit Sie Client-seitige Cascading Stylesheets (CSS) oder XSLT-Stylesheets mit Ihrem XML-Dokument verwenden können. Der Wert des Parameters –stylehref muss ein absoluter Pfad sein. Das Stylesheet kann einen beliebigen Namen haben, aber die Erweiterung muss .css oder .xsl sein. Weitere Informationen finden Sie unter „Verwenden von Server-seitiger und Client-seitiger Stylesheet-Verarbeitung“ auf Seite 41. Dieser Parameter wird in Verbindung mit dem Parameter –styletype verwendet.

Optional für: Alle Query-Befehle

Erfordert: –styletype-Parameter

Beispiel (setzt voraus, dass sich mystylesheet.xsl im Root-Ordner der Web-Server-Software befindet):

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–styletype=text/xsl
&–stylehref=/mystylesheet.xsl&–findall

Query-Parameter –styletype (Stilart)

Generiert eine Verarbeitungsanweisung für XML-Stylesheets im Ausgabedokument. Dabei setzt er den Wert des Typattributs (type=text/css oder type=text/xsl), damit Sie Cascading Stylesheets (CSS) oder XSLT-Dokumente mit Ihrem XML-Dokument verwenden können. Weitere Informationen finden Sie unter „Verwenden von Server-seitiger und Client-seitiger Stylesheet-Verarbeitung“ auf Seite 41. Dieser Parameter wird in Verbindung mit dem Parameter –stylehref verwendet.

Optional für: Alle Query-Befehle

Erfordert: –stylehref-Parameter

Beispiel (setzt voraus, dass sich mystylesheet.css im Root-Ordner der Web-Server-Software befindet):

http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–styletype=text/css&–stylehref=/mystylesheet.css&–findall

Query-Parameter –token.[string] (Werte zwischen XSLT-Stylesheets übergeben)

Übergibt beliebige benutzerdefinierte Informationen zwischen XSLT-Stylesheets, ohne Sitzungen oder Cookies zu verwenden. Dieser Query-Parameter kann nur für Custom Web Publishing mit XSLT-Abfragen verwendet werden.

***string* in –token.[*string*] ist:** Alphanumerischer String beliebiger Länge, außer Leerzeichen, einschließlich der Ziffern 0-9, Kleinbuchstaben a-z oder Großbuchstaben A-Z.

Benutzerdefinierter Parameterwert ist: Beliebige Zeichenfolge, die URL-kodiert ist.

Optional für: Alle XSLT-Abfragen

Beispiel:

http://192.168.123.101/fmi/xsl/template/my_stylesheet.xsl?–db=employees&–lay=departments&–grammar=fmresultset&–token.D100=Active&–findall

Weitere Informationen finden Sie unter „Verwendung von Tokens zur Übergabe von Informationen zwischen Stylesheets“ auf Seite 59.

Anhang B
Fehlercodes für Custom Web Publishing

Die Web Publishing Engine unterstützt drei Arten von Fehlercodes, die für Custom Web Publishing auftreten können:

- **Datenbank- und Query-String-Fehler**. Die Web Publishing Engine generiert für eine veröffentlichte Datenbank jedes Mal einen Fehlercode, wenn eine XML-Datenabfrage auftritt. Weitere Informationen finden Sie im nächsten Abschnitt, „Fehlercodenummern für FileMaker-Datenbanken“.
- **Web Publishing Engine-Fehler**. Wenn sich die Web Publishing Engine im Entwicklungsmodus befindet, generiert sie eine spezifische Fehlerseite, falls in der Web Publishing Engine selbst ein Fehler auftritt. Im Produktionsmodus wird eine allgemeine Textmeldung angezeigt. Siehe „Fehlercodenummern für die Web Publishing Engine“ auf Seite 114.
- **Fehler in FileMaker XSLT-Erweiterungsfunktionen.** Sie können die Erweiterungsfunktion fmxslt:check_error_status() in einem XSLT-Stylesheet verwenden, um den Fehlerstatus von Erweiterungsfunktionen bei ihrem Aufruf zu überprüfen. Siehe „Fehlercodenummern für die FileMaker XSLT-Erweiterungsfunktionen“ auf Seite 115.

Eine Liste der neuesten Fehlercodes finden Sie in der FileMaker Knowledge Base (www.filemaker.com/kb).

Fehlercodenummern für FileMaker-Datenbanken

Die Web Publishing Engine generiert für eine in XML-Format veröffentlichte Datenbank jedes Mal einen Fehlercode, wenn Daten angefordert werden. Diese Art von Fehlercodewert wird am Beginn des XML-Dokuments in das <error code>-Element für die fmresultset-Grammatik bzw. im <ERRORCODE>-Element für die Grammatik FMPXMLRESULT oder FMPDSORESULT eingefügt. Der Fehlercode 0 gibt an, dass kein Fehler aufgetreten ist.

Beispiel des Datenbankfehlercodes in der fmresultset-Grammatik:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE fmresultset PUBLIC "-//FMI//DTD fmresultset//EN" "/fmi/xml/fmresultset.dtd">
<fmresultset xmlns="http://www.filemaker.com/xml/fmresultset" version="1.0">
   <error code="0"></error>
```

Beispiel des Datenbankfehlercodes in der FMPXMLRESULT-Grammatik:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE FMPXMLRESULT PUBLIC "-//FMI//DTD FMPXMLRESULT//EN" "/fmi/xml/FMPXMLRESULT.dtd">
<fmpxmlresult xmlns="http://www.filemaker.com/fmpxmlresult">
   <ERRORCODE>0</ERRORCODE>
```

Als Entwickler der Custom Web Publishing-Lösung bleibt es Ihnen überlassen, den Wert des <error code>- bzw. <ERRORCODE>-Elements zu prüfen und geeignete Maßnahmen vorzusehen. Die Web Publishing Engine behandelt keine Datenbankfehler.

Fehlernummer	Beschreibung
-1	Unbekannter Fehler
0	Kein Fehler
1	Aktion durch Benutzer abgebrochen
2	Speicherfehler
3	Befehl nicht verfügbar (z. B. falsches Betriebssystem, falscher Modus etc.)
4	Befehl unbekannt.
5	Befehl ungültig (z. B. ist für den Scriptschritt „Feldwert setzen“ keine Formel angegeben).
6	Datei ist schreibgeschützt.
7	Speicherüberlauf
8	Leeres Ergebnis
9	Ungenügende Berechtigungen
10	Angeforderte Daten fehlen.
11	Name ist nicht gültig.
12	Name existiert bereits.
13	Datei oder Objekt ist in Gebrauch.
14	Außerhalb des gültigen Bereichs
15	Teilen durch null nicht möglich.
16	Operation fehlgeschlagen, Anfrage wiederholen (z. B. eine Benutzeranfrage).
17	Konvertierung von fremdem Zeichensatz in UTF-16 fehlgeschlagen.
18	Client muss Kontoinformationen liefern, um fortzufahren.
19	Zeichenfolge enthält andere Zeichen als A-Z, a-z, 0-9 (ASCII).
100	Datei fehlt.
101	Datensatz fehlt.
102	Feld fehlt.
103	Beziehung fehlt.
104	Script fehlt.
105	Layout fehlt.
106	Tabelle fehlt.
107	Index fehlt.
108	Werteliste nicht vorhanden.
109	Berechtigung fehlt.
110	Bezugstabellen fehlen.
111	Feldwiederholung ist ungültig.
112	Fenster fehlt.
113	Funktion fehlt.
114	Dateiverweis fehlt.

Fehlernummer	Beschreibung
115	Menüset fehlt.
116	Layoutobjekt fehlt.
117	Datenquelle fehlt.
130	Dateien sind beschädigt oder fehlen und müssen neu installiert werden.
131	Dateien des Sprachpakets fehlen (z. B. Vorlagendateien).
200	Zugriff auf Datensatz verweigert.
201	Feld kann nicht geändert werden.
202	Zugriff auf Feld verweigert.
203	Keine zu druckenden Datensätze in der Datei bzw. Passwort erlaubt kein Drucken.
204	Kein Zugriff auf Feld(er) in Sortierfolge
205	Benutzer hat keine Zugriffsrechte, um neue Datensätze zu erstellen; Import überschreibt bestehende Daten.
206	Benutzer hat keine Zugriffsrechte, um das Passwort zu ändern, oder die Datei ist schreibgeschützt.
207	Benutzer hat nicht genügend Zugriffsrechte, um das Datenbankschema zu ändern, oder die Datei ist schreibgeschützt.
208	Passwort enthält zu wenige Zeichen.
209	Neues Passwort muss sich vom bestehenden unterscheiden.
210	Benutzerkonto ist inaktiv.
211	Passwort ist abgelaufen.
212	Ungültiges Benutzerkonto und/oder Passwort. Versuchen Sie es erneut.
213	Benutzerkonto und/oder Passwort existieren nicht.
214	Zu viele Anmeldeversuche
215	Administratorrechte können nicht dupliziert werden.
216	Gastkonto kann nicht dupliziert werden.
217	Benutzer hat nicht genügend Zugriffsrechte, um Administratorkonto zu ändern.
300	Datei ist geschützt oder in Gebrauch.
301	Datei ist blockiert durch anderen Anwender.
302	Tabelle ist blockiert durch anderen Anwender.
303	Datenbankschema ist blockiert durch anderen Anwender.
304	Layout ist blockiert durch anderen Anwender.
306	Datensatzänderungs-ID stimmt nicht überein.
400	Suchkriterien sind leer.
401	Kein Datensatz entspricht der Abfrage.
402	Kein Abgleichsfeld für eine Referenz
403	Maximales Datensatzlimit für FileMaker Pro-Demo wird überschritten.
404	Ungültige Sortierfolge
405	Angegebene Datensatzzahl übersteigt die Anzahl der ausschließbaren Datensätze.

Fehlernummer	Beschreibung
406	Ungültige Kriterien für Ersetzen/Neunummerierung
407	Ein oder beide Gruppierfeld(er) fehlen (ungültige Beziehung).
408	Angegebenes Feld hat ein Datenformat, das diesem Befehl nicht entspricht.
409	Ungültige Importfolge
410	Ungültige Exportfolge
412	Falsche Version von FileMaker Pro verwendet, um die Datei wiederherzustellen
413	Angegebenes Feld hat ungeeigneten Feldtyp.
414	Layout kann das Ergebnis nicht anzeigen.
415	Ein oder mehrere erforderliche Bezugsdatensätze sind nicht verfügbar.
416	Primärschlüssel ist für Datenquelltabelle erforderlich.
417	Die Datenbank ist keine unterstützte Datenquelle.
500	Datumswert entspricht nicht den Überprüfungskriterien.
501	Zeitwert entspricht nicht den Überprüfungskriterien.
502	Zahlenwert entspricht nicht den Überprüfungskriterien.
503	Feldwert entspricht nicht der Bereichsüberprüfung.
504	Feldwert entspricht nicht der Eindeutigkeitsüberprüfung.
505	Feldwert existiert nicht in der Datenbank und entspricht nicht der Existenzüberprüfung.
506	Feldwert entspricht nicht der Überprüfung nach Bestandteil einer Werteliste.
507	Feldwert entspricht nicht der Überprüfung durch Berechnung.
508	Ungültiger Wert wurde in Suchenmodus eingegeben.
509	Feld verlangt gültigen Wert.
510	Bezugswert ist leer oder nicht verfügbar.
511	Der Wert im Feld überschreitet die maximale Anzahl der zulässigen Zeichen.
512	Der Datensatz wurde bereits durch anderen Anwender geändert.
513	Um einen Datensatz zu erstellen, muss der Datensatz in mindestens einem Feld einen Wert haben.
600	Druckerfehler aufgetreten.
601	Kombination von Kopf- und Fußbereich übersteigt eine Seitenlänge.
602	Datenbereich passt für aktuelle Spalteneinstellung nicht auf eine Seite.
603	Verbindung zum Drucker getrennt.
700	Datei hat falschen Dateityp für Import.
706	EPSF-Datei hat keine Übersichtsgrafik.
707	Grafikfilter nicht vorhanden.
708	Dateiimport nicht möglich bzw. Farbmonitor für Import erforderlich.
709	Import des QuickTime-Films fehlgeschlagen.
710	QuickTime-Dateiverweis konnte nicht aktualisiert werden, da für Datenbankdatei nur Lesezugriff besteht.

Fehlernummer	Beschreibung
711	Importfilter nicht vorhanden.
714	Zugriffsrechte reichen für diesen Befehl nicht aus.
715	Benannter Bereich oder Tabellenblatt von Excel konnte nicht gefunden werden.
716	Eine SQL Anfrage mit DELETE, INSERT oder UPDATE ist nicht für ODBC Import erlaubt.
717	Zum Fortsetzen des Imports bzw. Exports sind nicht genügend XML/XSL-Daten vorhanden.
718	XML-Parsingfehler (von Xerces)
719	Fehler beim Transformieren von XML mit XSL (von Xalan)
720	Fehler beim Export: Das vorgesehene Format unterstützt keine Wiederholfelder.
721	Im Parser oder Transformer ist ein unbekannter Fehler aufgetreten.
722	Daten können nicht in eine Datei importiert werden, die keine Felder hat.
723	Sie sind nicht berechtigt, Datensätze in der Zieltabelle hinzuzufügen oder zu ändern.
724	Sie sind nicht berechtigt, Datensätze in der Zieltabelle hinzuzufügen.
725	Sie sind nicht berechtigt, Datensätze in der Zieltabelle zu ändern.
726	In der Importdatei sind mehr Datensätze vorhanden als in der Zieltabelle. Nicht alle Datensätze werden importiert.
727	In der Zieltabelle sind mehr Datensätze vorhanden als in der Importdatei. Nicht alle Datensätze werden aktualisiert.
729	Fehler beim Import. Datensätze konnten nicht importiert werden.
730	Nicht unterstützte Excel-Version. Konvertieren Sie die Datei in Excel 7.0 (Excel 95), Excel 97, 2000 oder XP und versuchen Sie es erneut.
731	Die für den Import ausgewählte Datei enthält keine Daten.
732	Diese Datei kann nicht eingefügt werden, weil sie selbst weitere Dateien enthält.
733	Eine Tabelle kann nicht in sich selbst importiert werden.
734	Dieser Dateityp kann nicht als Bild dargestellt werden.
735	Dieser Dateityp kann nicht als Bild dargestellt werden. Er wird eingefügt und dann als Datei dargestellt.
736	Zu viele Daten für den Export in dieses Format. Daten werden abgeschnitten.
800	Datei konnte auf Datenträger nicht erstellt werden.
801	Temporärdatei konnte auf Systemdatenträger nicht erstellt werden.
802	Datei konnte nicht geöffnet werden. Dieser Fehler kann durch eine oder mehrere der folgenden Möglichkeiten verursacht werden: ▪ Ungültiger Datenbankame. ▪ Datei ist in FileMaker Server geschlossen. ▪ Ungültige Berechtigung.
803	Datei in Einzelbenutzer-Status oder Host nicht vorhanden.
804	Datei konnte in ihrem aktuellen Status nicht mit Nur-Lese-Zugriff geöffnet werden.
805	Datei ist beschädigt; stellen Sie die Datei wieder her.
806	Datei kann mit dieser Version von FileMaker Pro nicht geöffnet werden.

Fehlernummer	Beschreibung
807	Datei ist keine FileMaker Pro-Datei oder ist schwer beschädigt.
808	Datei kann wegen beschädigter Zugriffsrechte nicht geöffnet werden.
809	Datenträger voll
810	Datenträger fixiert
811	Temporärdatei kann nicht als FileMaker Pro-Datei geöffnet werden.
813	Fehler bei Datensatz-Synchronisation im Netzwerk
814	Datei(en) kann (können) nicht geöffnet werden, da die maximale Anzahl an Dateien geöffnet ist.
815	Referenzdatei konnte nicht geöffnet werden.
816	Datei konnte nicht konvertiert werden.
817	Der Bindungsschlüssel der Datei stimmt nicht mit dieser Laufzeitanwendung überein.
819	Lokale Kopie einer remote Datei kann nicht gespeichert werden.
820	Datei wird geschlossen.
821	Host hat die Verbindung getrennt.
822	FMI-Dateien nicht gefunden; fehlende Dateien neu installieren.
823	Datei kann nicht auf Einzelbenutzer gesetzt werden, da Gäste verbunden sind.
824	Datei ist beschädigt oder keine FileMaker-Datei.
900	Allgemeiner Fehler in der Rechtschreibprüfung
901	Standardwörterbuch nicht installiert.
902	Hilfe-System konnte nicht gestartet werden.
903	Befehl kann nicht in einer gemeinsam genutzten Datei verwendet werden.
904	Befehl kann nur in einer Datei verwendet werden, die von FileMaker Pro Server freigegeben wurde.
905	Kein aktives Feld ausgewählt, Befehl kann nur mit aktivem Feld verwendet werden.
906	Die aktuelle Datei ist nicht freigegeben. Der Befehl kann nur verwendet werden, wenn die Datei freigegeben ist.
920	Initialisierung der Rechtschreibprüfung nicht möglich.
921	Anwenderwörterbuch kann nicht zur Bearbeitung geladen werden.
922	Anwenderwörterbuch existiert nicht.
923	Anwenderwörterbuch ist schreibgeschützt.
951	Ein unerwarteter Fehler ist aufgetreten.
954	Nicht unterstützte XML-Grammatik
955	Kein Datenbankname
956	Maximale Anzahl von Datenbanksitzungen überschritten.
957	Widersprüchliche Befehle
958	Parameter fehlt in Query
1200	Generischer Rechenfehler
1201	In dieser Funktion gibt es zu wenige Parameter.

Fehlernummer	Beschreibung
1202	In dieser Funktion gibt es zu viele Parameter.
1203	Unerwartetes Ende der Berechnung
1204	Es wird eine Zahl, eine Textkonstante, ein Feldname oder „(“ erwartet.
1205	Kommentar ist nicht mit „*/“ beendet.
1206	Textkonstante muss mit einem Anführungszeichen enden.
1207	Klammer unvollständig
1208	Operator fehlt, Funktion nicht gefunden oder „(“ nicht erwartet.
1209	Name (z. B. Feldname oder Layoutname) fehlt.
1210	Plugin-Funktion wurde bereits registriert.
1211	In dieser Funktion ist die Listennutzung nicht zulässig.
1212	Hier wird ein Operator (z. B. +, -, *) erwartet.
1213	Diese Variable wurde bereits in der SetzeVars-Funktion definiert.
1214	MITTELWERT, ANZAHL, ERWEITERN, HOLEWIEDERHOLFELDWERT, MAX, MIN, NBW, STABW, SUMME und HOLEERGEBNISWERT: Ausdruck gefunden, wo nur ein Feld benötigt wird.
1215	Dieser Parameter ist ein ungültiger Statusfunktionsparameter.
1216	Als erstes Argument einer HOLEERGEBNISWERT-Funktion können nur Statistikfelder angegeben werden.
1217	Gruppierfeld ist ungültig.
1218	Zahl kann nicht berechnet werden.
1219	Ein Feld kann nicht in seiner eigenen Formel benutzt werden.
1220	Feldtyp muss normal oder berechnet sein.
1221	Datentyp muss Zahl, Datum, Zeit oder Zeitstempel sein.
1222	Formel kann nicht gespeichert werden.
1223	Die angegebene Funktion ist noch nicht implementiert.
1224	Die angegebene Funktion existiert nicht.
1225	Die angegebene Funktion wird in diesem Kontext nicht unterstützt.
1400	ODBC-Client-Treiberinitialisierung fehlgeschlagen. Stellen Sie sicher, dass die ODBC-Client-Treiber richtig installiert sind.
1401	Umgebung konnte nicht zugeteilt werden (ODBC).
1402	Umgebung konnte nicht freigegeben werden (ODBC).
1403	Verbindung trennen fehlerhaft (ODBC).
1404	Verbindung konnte nicht zugeteilt werden (ODBC).
1405	Verbindung konnte nicht freigegeben werden (ODBC).
1406	Überprüfung von SQL API fehlgeschlagen (ODBC).
1407	Anweisung konnte nicht zugeteilt werden (ODBC).
1408	Erweiterter Fehler (ODBC)

Fehlernummer	Beschreibung
1409	Erweiterter Fehler (ODBC)
1410	Erweiterter Fehler (ODBC)
1411	Erweiterter Fehler (ODBC)
1412	Erweiterter Fehler (ODBC)
1413	Erweiterter Fehler (ODBC)

Fehlercodenummern für die Web Publishing Engine

Wenn sich die Web Publishing Engine im Entwicklungsmodus befindet, generiert sie eine spezifische Fehlerseite, sobald in der Web Publishing Engine selbst ein Fehler auftritt. Dieser Fehlertyp kann eine Vielzahl von Ursachen haben, z. B. wenn die Web Publishing Engine zu Folgendem nicht in der Lage ist:

- Finden einer angeforderten oder verschachtelten (über <xsl:include>) Stylesheet-Datei
- Parsen einer angeforderten oder verschachtelten Stylesheet-Datei aufgrund eines XML-Fehlers in der Datei
- Generieren eines Stylesheets aus der Datei aufgrund eines XSLT- oder XPath-Fehlers in der Datei
- Verarbeiten der Abfrage, da die XML-Grammatik im CGI nicht korrekt angegeben ist
- Kommunizieren mit dem Web Publishing Core, um XML abzurufen

Wenn die Web Publishing Engine im Entwicklungsmodus läuft, enthält die Fehlerseite für diesen Fehlertyp eine Fehlermeldung und eine eingeklammerte Fehlernummer. Beispiel:

FileMaker Custom Web Publishing mit XSLT
Die abgefragte Stylesheet-Datei wurde nicht gefunden. (FILE-ER0001)
Datei: home4.xsl
Zeile:
Spalte:
Hinweis: Wenn eine Quelldatei und eine Zeilennummer angegeben sind, handelt es sich hierbei um die Stelle, an der der Fehler entdeckt wurde. Dies muss aber nicht die genaue Stelle des Problems sein.

Beispiel für Fehlerseite, wenn sich die Web Publishing Engine im Entwicklungsmodus befindet

Die folgende Liste führt einige der Web Publishing Engine-Fehlercodewerte auf:

Fehlercode-Wert	Beschreibung
QUERY-ER0001	Im Query-Parameter –grammar ist keine XML-Grammatik angegeben.
QUERY-ER0002	„xxx“ ist keine gültige XML-Grammatik für FileMaker XSLT.
FILE-ER0001	Die abgefragte Stylesheet-Datei wurde nicht gefunden.
FILE-ER0002	Die angeforderte Datei wurde nicht gefunden.
UNKNOWN	Ein unerwarteter Fehler ist aufgetreten.
MCS-000 bis MCS-600	Ein unerwarteter Fehler ist aufgetreten.

Fehlercode-Wert	Beschreibung
MCS-601	Die Ressource „x“ konnte nicht geladen werden, weil Ressourcen dieses Typs nicht unterstützt werden: „x“.
MCS-602	Der URL „x“ konnte nicht aufgelöst werden.
MCS-603	Die HTTP-Anfrage für „x“ gab einen Fehler des Typs „x“ zurück.
MCS-604	Die Ressource „x“ konnte nicht geladen werden, weil ein unerwarteter Fehler aufgetreten ist.
MCS-605	Die Ressource „x“ konnte nicht geladen werden, weil der Content-Typ ungültig war.
MCS-606	Die Ressource „x“ konnte nicht geladen werden, weil das Dokument einen XML-Fehler aufweist.
MCS-607	Die Ressource „x“ konnte aufgrund eines Authentifizierungsproblems nicht geladen werden.
MCS-700	Ein unerwarteter Fehler ist aufgetreten.
MCS-800	Ein unerwarteter Fehler ist aufgetreten.

Wenn sich die Web Publishing Engine im Produktionsmodus befindet, wird für Fehler in der Web Publishing Engine die folgende allgemeine Textmeldung auf der Seite pe_server_error.html angezeigt:

Bei der Verwendung von FileMaker Custom Web Publishing mit XSLT ist ein unerwarteter Fehler aufgetreten.

Die Standarddatei pe_server_error.html enthält die vorangehende Textmeldung in sechs Sprachen.

Als Entwickler können Sie den Text der Fehlerseite pe_server_error.html wie für Ihre Lösung erforderlich bearbeiten. Die Datei pe_server_error.html befindet sich im Ordner cwpe innerhalb des publishing-engine-Ordners auf dem Host, auf dem Sie die Web Publishing Engine installiert haben.

Informationen über das Setzen der Web Publishing Engine in den Entwicklungs- oder Produktionsmodus finden Sie in der FileMaker Server Hilfe.

Fehlercodenummern für die FileMaker XSLT-Erweiterungsfunktionen

Die Erweiterungsfunktion fmxslt:check_error_status() (siehe „Prüfen des Fehlerstatus von Erweiterungsfunktionen“ auf Seite 76) gibt einen der Fehler in der folgenden Tabelle zurück:

Fehlercode-Wert	Beschreibung
-1	Unbekannter Fehler
0	Kein Fehler
	Allgemeine Fehler
10000	Ungültiger Header-Name
10001	Ungültiger HTTP-Statuscode

Fehlercode-Wert	Beschreibung
	Sitzungsfehler
10100	Unbekannter Sitzungsfehler
10101	Angeforderter Sitzungsname bereits in Gebrauch.
10102	Zugriff auf Sitzung nicht möglich – sie existiert evtl. nicht.
10103	Zeitüberschreitung der Sitzung
10104	Angegebenes Sitzungsobjekt existiert nicht.
	Nachrichtenfehler
10200	Unbekannter Nachrichtenfehler
10201	Nachrichten-Formatfehler
10202	SMTP-Feldfehler in Nachricht
10203	Fehler in „An-Feld“ der Nachricht
10204	Fehler in „Von-Feld“ der Nachricht
10205	Fehler in „CC-Feld“ der Nachricht
10206	Fehler in „BCC-Feld“ der Nachricht
10207	Fehler in „Betreff-Feld“ der Nachricht
10208	Fehler in „Reply-To-Feld“ der Nachricht
10209	Nachrichten-Textfehler
10210	Rekursiver Mail-Fehler _ Aufruf von send_email() in E-Mail-XSLT-Stylesheet versucht
10211	SMTP-Authentifizierungsfehler – Anmeldung fehlgeschlagen oder falscher Authentifizierungstyp angegeben
10212	Unzulässige Funktionsverwendung – Aufruf von call set_header(), set_status_code() oder set_cookie() in E-Mail-XSLT-Stylesheet versucht
10213	SMTP-Server ist ungültig oder funktioniert nicht.
	Formatierungsfehler
10300	Unbekannter Formatierungsfehler
10301	Ungültiges Datum-Zeit-Format
10302	Ungültiges Datumsformat
10303	Ungültiges Zeitformat
10304	Ungültiges Tagesformat
10305	Falsch formatierter Datum-Zeit-String
10306	Falsch formatierter Datums-String
10307	Falsch formatierter Zeit-String
10308	Falsch formatierter Tages-String
10309	Nicht unterstützte URL-Textkodierung
10310	Ungültige URL-Textkodierung
10311	Musterfehler in regulärem Ausdruck

Index

A

Abfragen von Ausschnittfeldern 91
Abfragen von XML-Daten 27
Abrufen verfügbarer Scriptnamen 94
Admin Console für Web Publishing Engine 27
Administrationskonsole für Web Publishing Engine 45
Aktivieren von Instant Web Publishing in Datenbank 19
Angeben der XML-Grammatik 55
ASCII-Zeichen, in XML-Dokumenten 38
Attribut für automatische Eingabe 33
Attribut für vierstellige Jahreszahlen 33
Ausblenden von Metadaten mit Stylesheets 43
Ausgabe von Daten mit Stylesheets 43
Ausgabeseiten
  Anfängliche Standardkodierungseinstellung 58
  Ausgabemethode, angeben 58
  Kodierung, angeben 58
  <xsl:output>-Element 58
Ausschnitte
  Datensätze bearbeiten 90
  Datensätze hinzufügen 89
  Layouts 100
  Löschen von Datensätzen 90
  Sortieren von Datensätzen 100
Ausschnittfeld-Queries 100
authenticated base URI, verwenden des Parameters 62
Authentifizierung von Web-Benutzern 19
  Passwort 62
Authentifizierung Web-Benutzer 19, 61
  Basis 61

B

Basis-URI-Parameter 61
Befehle, Query *s.* Query-Strings
Beispiele für
  Generierte FMPXMLLAYOUT-Grammatik 37
  Generierte FMPXMLRESULT-Grammatik 36
  Generierte fmresultset-Grammatik 34
Benutzernamen
  Authentifizierung Web-Benutzer 19, 61
  Für Custom Web Publishing definieren 19
  Zugriff auf XML-Dokumente 62
Berechtigung, Zuweisen für Custom Web Publishing 19
Beschränken von Ausschnittfeldzeilen 100
break_encode(), Erweiterungsfunktion 70

C

check_error_status(), Erweiterungsfunktion 76, 115
Clientinformationen, über XSLT-Parameter abrufen 61
Client-seitige Stylesheets 27, 41
compare_date(), Erweiterungsfunktion 73
compare_datetime(), Erweiterungsfunktion 74
compare_day(), Erweiterungsfunktion 73
compare_time(), Erweiterungsfunktion 73
Compound-Suche (Query-Befehl) 93
Compound-Suche (Query-Parameter) 99
contains_checkbox_value(), Erweiterungsfunktion 71
convert_datetime(), Erweiterungsfunktion 74
Cookies
  Erweiterungsfunktionen, verwenden 69
  Sitzungs-ID speichern 64
create_session(), Erweiterungsfunktion 65
Custom Web Publishing
  Beschreibung 15, 16
  Definition 11
  Erweitertes Zugriffsrecht für 19
  Gastkonto 20
  In Datenbank aktivieren 19
  In Web Publishing Engine aktivieren 20
  Mit PHP 13
  Mit XML 13
  Mit XSLT 13, 44, 51
  Neue Funktionen in 16
  Scripts 24
  Scripts verwenden 22
  Statische IP-Adresse verwenden 18
  Übersicht 11
  Voraussetzungen für das 17
  Web-Benutzer-Zugriff auf Lösungen 61
  XML verwenden 25
  Zugriff auf Lösungen von Web-Benutzern 19
  Zugriff von IP-Adressen in Web-Server beschränken 21
Custom Web Publishing Engine (CWPE) 26, 44

D

Element <datasource> 32
Datenbanken, veröffentlichte schützen 20
Datenbankfehlercodes 31
Datenbanksitzungen, aktivieren 66, 91
Datums-Erweiterungsfunktionen, verwenden 72
Datumsformatstrings 74
–db (Query-Parameter) 95
–dbnames (Query-Befehl) 92
Definieren von Erweiterungsfunktionen 77
–delete (Query-Befehl) 92
–delete.related (Query-Parameter) 90
Document Type Definitions (DTDs) 31, 35
document(), Funktion 62
Dokumentation 9
Dokumente, über document()-Funktion laden 62
–dup (Query-Befehl) 92

E

–edit (Query-Befehl) 92
Elektronische Dokumentation 9
Element <metadata> 33
Elemente
  Datenbankfehlercodes 31
  In FMPXMLLAYOUT-Grammatik 36
  In FMPXMLRESULT-Grammatik 35
  In fmresultset-Grammatik 32
E-Mail-Nachrichten
  Anfängliche Standardkodierungseinstellung 58
  Erweiterungsfunktionen für 66
–encoding (Query-Parameter) 95
Entwicklungsmodus, Web Publishing Engine 114
Erneut anmelden, Script 20
<error code-> und <ERRORCODE>-Elemente 107
Erstellen eines neuen Datensatzes 94
Erweitertes Zugriffsrecht für Custom Web Publishing 19
Erweiterungsfunktionen für FileMaker XSLT
  Siehe auch fmxslt-Erweiterungsfunktionen
Exportieren von XML-Daten 25
Extensible Markup Language (XML) *siehe* XML

F

Fehler
  Datenbankfehlercode-Elemente 31
  Datenbankfehlercodes 107
  Erläuterung von Fehlercodes 107
  Erweiterungsfunktionen, Fehlercodenummern 115
  Fehlerstatus von Erweiterungsfunktionen prüfen 76, 115
  pe_application_log.txt (Protokolldatei) 84
  pe_server_error.html (Fehlerseite) 115
  Protokolldateien für Web-Server 84
  Web Publishing Engine-Fehlercodenummern 114
Fehlerbehebung
  Custom Web Publishing-Websites 82
  XML-Dokumentzugriff 42
  XSLT-Stylesheets 49
Feldname (Query-Parameter, nicht für Medienfeld) 96
Feldnamen, voll qualifizierte Syntax 89
–field (Query-Parameter, für Medienfeld) 95
<field-definition>-Element 33
–fieldname.op (Query-Parameter) 96
FileMaker API für PHP 13
  Definition 13
FileMaker Pro, Unterschied zu Web Publishing Engine 25
FileMaker Server
  Dokumentation 9
  Installieren 9
FileMaker Server Admin
  Siehe Admin Console 20
FileMaker Server Admin Console 27
FileMaker Site-Assistent *Siehe* XSLT-Site-Assistent
FileMaker-spezifische XSLT-Parameter 60
Filtern von Ausschnittfeldzeilen 100
Filtern von Daten mit Stylesheets 43
–find (Query-Befehl) 92
–findall (Query-Befehl) 92
–findany (Query-Befehl) 92
–findquery (Query-Befehl) 93
FMPDSORESULT-Grammatik
  Im Vergleich mit anderen Grammatiken 30
FMPXMLLAYOUT-Grammatik 25, 36–37
  Im Vergleich mit anderen Grammatiken 30
FMPXMLRESULT-Grammatik 25, 35–36
  Im Vergleich mit anderen Grammatiken 30
fmresultset-Grammatik 25, 32–34
  Im Vergleich mit anderen Grammatiken 30
fmxml (Schlüsselwort zur Aktivierung von XML-Publishing) 19, 27
fmxml (Schlüsselwort zur Aktivierung von XSLT-Publishing) 19, 45

fmxslt-Erweiterungsfunktionen
fmxslt:break_encode(), Funktion 70
fmxslt:check_error_status(), Funktion 76, 115
fmxslt:compare_date(), Funktion 73
fmxslt:compare_datetime(), Funktion 74
fmxslt:compare_day(), Funktion 73
fmxslt:compare_time(), Funktion 73
fmxslt:contains_checkbox_value(), Funktion 71
fmxslt:convert_datetime(), Funktion 74
fmxslt:create_session(), Funktion 65
fmxslt:get_cookie(), Funktion 69
fmxslt:get_cookies(), Funktion 69
fmxslt:get_date(), Funktion 72
fmxslt:get_datetime(), Funktion 74
fmxslt:get_day(), Funktion 73
fmxslt:get_fm_date_format(), Funktion 73
fmxslt:get_fm_date_format()-Funktion 73
fmxslt:get_fm_timestamp_format(), Funktion 73
fmxslt:get_header(), Funktion 68
fmxslt:get_long_date_format(), Funktion 73
fmxslt:get_long_day_format(), Funktion 73
fmxslt:get_long_time_format(), Funktion 73
fmxslt:get_session_object(), Funktion 65
fmxslt:get_short_date_format(), Funktion 73
fmxslt:get_short_day_format(), Funktion 73
fmxslt:get_short_time_format(), Funktion 73
fmxslt:get_time(), Funktion 72, 73
fmxslt:html_encode(), Funktion 70
fmxslt:invalidate_session(), Funktion 65, 66
fmxslt:regex_contains(), Funktion 71
fmxslt:remove_session_object(), Funktion 65
fmxslt:send_email(), Funktionen 66, 67
fmxslt:session_encode_url(), Funktion 64, 65
fmxslt:session_exists(), Funktion 65
fmxslt:set_cookie(), Funktion 69
fmxslt:set_header(), Funktion 68
fmxslt:set_session_object(), Funktion 65
fmxslt:set_session_timeout (), Funktion 65
fmxslt:set_status_code(), Funktion 68
fmxslt:url_decode(), Funktion 70
fmxslt:url_encode(), Funktion 70
Formatieren von Daten mit Stylesheets 43
Formatstrings, Datum und Zeit 74

G

Gastkonto
Aktivieren 20
Deaktivieren 20
Mit Custom Web Publishing 20
Generieren einer statischen Seite 58
get_cookie(), Erweiterungsfunktion 69
get_cookies(), Erweiterungsfunktion 69
get_date(), Erweiterungsfunktion 72
get_datetime(), Erweiterungsfunktion 74
get_day(), Erweiterungsfunktion 73
get_fm_date_format(), Erweiterungsfunktion 73
get_fm_time_format(), Erweiterungsfunktion 73
get_fm_timestamp_format(),
Erweiterungsfunktion 73
get_header(), Erweiterungsfunktion 68
get_long_date_format(), Erweiterungsfunktion 73
get_long_day_format(), Erweiterungsfunktion 73
get_long_time_format(), Erweiterungsfunktion 73
get_session_object(), Erweiterungsfunktion 65
get_short_date_format(), Erweiterungsfunktion 73
get_short_day_format(), Erweiterungsfunktion 73
get_short_time_format(), Erweiterungsfunktion 73
get_time(), Erweiterungsfunktion 72, 73
GIF-Dateien, im Web veröffentlichen 22
Globales Attribut 33
–grammar (Query-Parameter) 55, 97
Grammatik, für XSLT empfohlen 55
Grammatiken für XML, Beschreibung 30

H

Header-Funktionen, verwenden 68
Hinweise zur Dokumentation 9, 18
HTML
Formulare für XML-Abfragen 27
XML-Daten umformatieren in 25
html_encode(), Erweiterungsfunktion 70

I

Importieren von XML-Daten 25
Informationen zwischen Stylesheets übergeben 59
Inhaltspufferung, verwenden 63
Installationsdokumentation 9
Instant Web Publishing
Definition 11
Dokumentation 9

Integrieren von Daten mit Stylesheets 43
invalidate_session(), Erweiterungsfunktion 65, 66
ISO-2022-JP-Kodierung 57
ISO-8859-15-Kodierung 57
ISO-8859-1-Kodierung 57

J

JavaScript
  Definieren von Erweiterungsfunktionen 77
JDBC-Dokumentation 9
JPEG-Dateien, im Web veröffentlichen 22
jsessionid-Parameter 64

K

Kodierung
  Aufträge 57
  Ausgabe über Element <xsl:output> 58
  –encoding (Query-Parameter) 57, 95
  Erweiterungsfunktionen zur Stringmanipulation verwenden 70
  URL 29, 64
  XML-Daten 30, 38
  XSLT-Stylesheets 58
Konten und Zugriffsrechte
  Für Custom Web Publishing aktivieren 19
  Gastkonto 20
  Scripts 23
Kontrollkästchen, Werte prüfen in 71

L

Laden zusätzlicher Dokumente 62
–lay (Query-Parameter) 40, 97
–lay.response (Query-Parameter) 40, 98
Layout bei Abfrage von XML-Daten angeben 40
Layoutinformation, in Stylesheet verwenden 63
Layoutinformationen abrufen 94
Layoutnamen abrufen 93
–layoutnames (Query-Befehl) 93
Layouts, für eine XML-Antwort wechseln 40
Logs (Ordner) 84
–lop (Query-Parameter) 98
Löschen von Ausschnittdatensätzen 90

M

Mail-Nachrichten *siehe* E-Mail-Nachrichten
–max (Query-Parameter) 98
max-characters, Attribut 33
max-repeat, Attribut 33
Medienfelder
  Inhalt veröffentlichen 21, 49
  URL-Syntax für Zugriff in XML-Lösungen 28
  URL-Syntax für Zugriff in XSLT-Lösungen 53
  Wie Web-Benutzer auf Daten zugreifen 22
Metadaten, mit Stylesheets ausblenden 43
method-Attribut, <xsl:output>-Element 58
MIME (Multipurpose Internet Mail Extensions)-Typen 21
–modid (Query-Parameter) 98

N

name, Attribut 33
Namespaces für
  XML 31
  XSLT 55
Neue Funktionen in Custom Web Publishing 16
–new (Query-Befehl) 94
not-empty, Attribut 33
numeric-only, Attribut 33
Nummern für
  Datenbankfehlercodes 107
  Erweiterungsfunktions-Fehlercodes 115
  Web Publishing Engine-Fehlercodes 114

O

ODBC-Dokumentation 9
Online-Dokumentation 9
Operatoren, Vergleich 96

P

Parameter für XSLT, FileMaker-spezifisch 60
Parameter, Query *s.* Query-Strings
Passwort ändern (Script) 20
Passwörter
  Authentifizierung Web-Benutzer 19, 61
  Für Custom Web Publishing definieren 19
  Kein Anmeldepasswort 20
  Passwort ändern (Script) 20
  Zugriff auf XML-Dokumente 62
PDFs 9
pe_application_log.txt (Protokolldatei) 84
pe_internal_access_log.txt (Protokolldatei) 85
pe_server_error.html (Fehlerseite) 115
PHP
  Fehlerbehebung 49
  Vorteile 14
PHP API für Custom Web Publishing 13

PHP-Publishing testen 49
–process (Query-Befehl) 58, 94
Produktionsmodus, Web Publishing Engine 115
Programmprotokoll 76, 84
Protokolldateien 82, 85
  Beschreibung 83
  pe_application_log.txt 84
  pe_internal_access_log.txt 85
  Protokollierung über <xsl:message>-Element 76
  web_server_module_log.txt 85
  Web-Server-Zugriff 84
  <xsl:message>-Element 84
Pufferung, in Stylesheet verwenden 63

Q

–query (Query-Parameter) 99
Query-Information, Zugriff in Abfrage 60
Query-String-Referenz 89
Query-Strings 38, 54, 87
  Befehle und Parameter 38, 54, 87
  Datensätze in Ausschnitte einfügen 89
  Datensätze in Ausschnitten bearbeiten 90
  Query-String-Referenz 89
  Richtlinien für 88
  Statisch definiert in XSLT-Stylesheets 56
  Variablenfeld, Syntax 91
  Voll qualifizierter Feldname, Syntax 89
  XML 38
  XML-Daten abfragen 87
  XSLT-Stylesheets, verwenden in 54
QuickTime-Filme, im Web veröffentlichen 22

R

–recid (Query-Parameter) 100
regex_contains(), Erweiterungsfunktion 71
Reguläre Perl-Ausdrücke, Vergleich von Strings 71
Reihenfolge der XML-Abfrageverarbeitung 41
<relatedset-definition>-Element 33
–relatedsets.filter (Query-Parameter) 100
–relatedsets.max (Query-Parameter) 100
remove_session_object(), Erweiterungsfunktion 65
result, Attribut 33
<resultset>-Element 33

S

SAT
  Siehe Admin Console 20
Scalable Vector Graphics (SVG), Umwandeln von XML-Daten in 25
Schlüsselwörter zur Aktivierung von Custom Web Publishing 19, 27, 45
Schützen veröffentlichter Datenbanken 20
–script (Query-Parameter) 101
–script.param (Query-Parameter) 101
–script.prefind (Query-Parameter) 101
–script.prefind.param (Query-Parameter) 102
–script.presort (Query-Parameter) 102
–script.presort.param (Query-Parameter) 102
–scriptnames (Query-Befehl) 94
Scripts
  Datenbanksitzungen, aktivieren 66
  Erneut anmelden 20
  Für XML-Abfragen 27
  In Custom Web Publishing 22
  Konten und Zugriffsrechte 23
  Passwort ändern 20
  Tipps und Überlegungen 23
send_email(), Erweiterungsfunktionen 66
Server-seitige XSLT-Stylesheets 43, 51
session_encode_url(), Erweiterungsfunktion 64
session_exists(), Erweiterungsfunktion 65
set_cookie(), Erweiterungsfunktion 69
set_header(), Erweiterungsfunktion 68
set_session_object(), Erweiterungsfunktion 65
set_session_timeout (), Erweiterungsfunktion 65
set_status_code(), Erweiterungsfunktion 68
Shift_JIS-Kodierung 57
Sicherheit
  Dokumentation 12
  Konten und Passwörter 21
  Richtlinien zum Schützen veröffentlichter Datenbanken 20
  Statisch definierte Query-Strings, verwenden 56
  Zugriff von IP-Adressen beschränken 21
Site-Assistent
  Beschreibung 16
Site-Assistent *Siehe* XSLT-Site-Assistent

Sitzungserweiterungsfunktionen, in Stylesheets verwenden 64, 65
–skip (Query-Parameter) 103
–sortfield (Query-Parameter) 103
–sortorder (Query-Parameter) 103
Speichern von Informationen in Sitzungen 64, 65
SSL (Secure Sockets Layer)-Verschlüsselung 21
Statisch definierte Query-Strings in XSLT-Stylesheets 56
Statisches Veröffentlichen, Definition 11
Status, in Sitzungen speichern 64, 65
Strings
  Erweiterungsfunktionen zur Stringmanipulation verwenden 70
  Vergleich über reguläre Perl-Ausdrücke 71
–stylehref (Query-Parameter) 104
Stylesheets
  Anweisung zur Verarbeitung von XML-Stylesheets 41
  Beispiele für Verwendung 43
  Client-seitig 41
  Cookie-Erweiterungsfunktionen 69
  Datums- und Zeitformatstrings 74
  Datums-, Zeit- und Tages-Erweiterungsfunktionen 72
  E-Mail-Nachrichten, senden 66
  Entwickeln 51
  Erläuterung 43
  Erstellen mit XSLT-Site-Assistent 46
  Erweiterungsfunktionen zur Stringmanipulation 70
  Fehlerstatus von Erweiterungsfunktionen, prüfen 76
  –grammar-Parameter für 55
  Header-Funktionen, verwenden 68
  Inhaltspufferung, verwenden 63
  Kodierung von 58
  Kontrollkästchen, Werte prüfen in 71
  Layoutinformation verwenden in 63
  Query-Strings für 54
  Richtlinien für Entwicklung 51
  Server-seitig 43, 51
  Sitzungsfunktion, verwenden 64, 65
  Testen 82, 83
  Vergleich von Strings über reguläre Perl-Ausdrücke 71
  Verwenden in Website oder Programm 48
  XSLT, Beschreibung 43
–styletype (Query-Parameter) 104

T

Tages-Erweiterungsfunktionen, verwenden 72
Technologietests 49
Testen
  Websites 82
  XML-Ausgabe 83
Textkodierung
  Anfängliche Standardeinstellungen 58
  –encoding (Query-Parameter) 57, 95
  Erweiterungsfunktionen zur Stringmanipulation verwenden 70
  Für XSLT-Abfragen 57
  Generierte XML-Daten 30
  Kodierungseinstellungen 57
  Standard für Abfragen und Ausgabeseiten 57
  URL 29, 64
time-of-day, Attribut 33
–token (Query-Parameter) 59, 105
Tools für XSLT, Beschreibung 16, 46
type, Attribut 33

U

Überblick über Schritte für
  XML-Datenzugriff 27
  XSLT Publishing 45
Übersicht
  Custom Web Publishing 11
Überwachen von Websites 83
Umwandeln von Daten mit Stylesheets 43
Unicode-Zeichen 38
url_decode(), Erweiterungsfunktion 70
url_encode(), Erweiterungsfunktion 70
URL-Syntax für
  Medienobjekte in XML-Lösungen 28
  Medienobjekte in XSLT-Lösungen 53
  XML-Abfragen 28
  XSLT-Stylesheets 52
URL-Textkodierung 29
US-ASCII-Kodierung 57
User-Agent-Header, prüfen 59
UTF-8 (Unicode Transformation 8 Bit)
  Format 29, 38
  Kodierungseinstellung 57

V

Variablenfelder
Datenbanksitzungen, aktivieren 66, 91
Mit Sitzungen verwenden 66, 91
Syntax 91
vCards, Umformatieren von XML-Daten in 25
Verarbeiten von XSLT-Stylesheets 94
Verarbeitung einer Web Publishing Engine-Anforderung 12
Verfügbare Datenbanklayouts 93
Verfügbare Scripts 94
Vergleich von Strings 71
Vergleichsoperatoren für Felder 96
Veröffentlichen im Web
Datenbankfehlercodes 107
Medienfeldobjekte 21, 49
Mit XSLT 16, 45, 51
QuickTime-Filme 22
Schützen von Datenbanken 20
Verbindung zum Internet oder Intranet 18
Voraussetzungen für das 17
XML verwenden 15, 27
–view (Query-Befehl) 94
Voll qualifizierter Feldname, Syntax 89
Voraussetzungen für Custom Web Publishing 17
Vorteile von XML und XSLT 14

W

Web Publishing Core
Illustration 26, 44
Interne Zugriffsprotokolle 85
Web Publishing Engine
Administrationskonsole 45
Admin Console 27
Anforderungsverarbeitung 12
Beschreibung 12
Entwicklungsmodus 114
Erzeugte Fehlercodes 107
Produktionsmodus 115
Programmprotokoll 84
Seiten aus XSLT-Stylesheet generieren 44
Vorteile 15
XML-Daten generieren 26
XML-Dokumente erzeugen 27
web_server_module_log.txt (Protokolldatei) 85
Web-Benutzer
Medienfelddaten verwenden 22
Voraussetzungen für Zugriff auf Custom Web Publishing-Lösungen 18
Zugriff auf geschützte Datenbanken 19, 62
Zugriff, geschützte Datenbanken 61
Webbrowser
Ausgabe empfangen 12
Rolle in XML-Abfragen 26
Rolle in XSLT-CWP-Abfragen 44
Web-Ordner, Medienfeldobjekte kopieren 22
Web-Server
MIME-Typenunterstützung 21
Protokolldateien 84
Rolle in XML-Abfragen 26
Rolle in XSLT-CWP-Abfragen 44
Websites
FileMaker Support-Seiten 9
Mit FileMaker Web Publishing Engine erstellen 15
Testen 82
Überwachen 83
Wechseln des Layouts für eine XML-Antwort 40
Werte, prüfen in Kontrollkästchen 71
wpc_access_log.txt (Datei) 85

X

XML
Anweisung zur Verarbeitung von XML-Stylesheets 41
Beschreibung 25
Client-seitige Stylesheets verwenden 41
Daten abfragen 27
Daten filtern 25
Document Type Definitions (DTDs) 31, 32, 35
Fehlerbehebung beim Zugriff auf XML-Dokumente 42
FMPXMLLAYOUT-Grammatik 36
FMPXMLRESULT-Grammatik 35
fmresultset-Grammatik 32
<field-definition> Element 33
<relatedset-definition> Element 33
Element <datasource> 32
Element <metadata> 33
Element <resultset> 33
Grammatiken, Beschreibung 30

Im UTF-8-Format kodiert 30, 38
In Datenbank aktivieren 19
Namespaces für 31
Parser 27, 38
Query-Strings 38, 87
Reihenfolge der Abfrageverarbeitung 41
URL-Textkodierung 29
XML 1.0-Spezifikation 25
XML-Daten aus Abfragen generieren 26
Zusammenfassung der Schritte für Zugriff auf XML-Daten 27
XML Custom Web Publishing 13
XML-Abfrage
Layout angeben 40
XML-Antwort
Layout wechseln 40
XPath-Anweisungen 59
<xsl:stylesheet>-Element 55, 59, 60, 83
<xsl:message>-Element 76
<xsl:output>-Element 58
<xsl:param name="authenticated-xml-base-uri"/>-Parameter 62
<xsl:param name="client-ip"/>-Parameter 61
<xsl:param name="client-password"/>-Parameter 61
<xsl:param name="client-user-name"/>-Parameter 61
<xsl:param name="request-query"/>-Parameter 60
<xsl:param name="xml-base-uri"/>-Parameter 61
<xsl:param>-Element 60
<xsl:template>-Element 61, 62, 83
<xsl:variable>-Element 62
XSLT
Beispiele für Stylesheets 43
Beschreibung 43
Cookie-Erweiterungsfunktionen 69
Datums- und Zeitformatstrings 74
Datums-, Zeit- und Tages-Erweiterungsfunktionen 72
E-Mail-Nachrichten, senden 66
Entwickeln von XSLT-Stylesheets 51
Erweiterungsfunktionen für FileMaker 59
Erweiterungsfunktionen zur Stringmanipulation 70
Fehlerbehebung in Stylesheets 49
Fehlerstatus von Erweiterungsfunktionen, prüfen 76
FileMaker-spezifische XSLT-Parameter 60
–grammar-Parameter 55
Header-Funktionen, verwenden 68
In Datenbank aktivieren 19
Inhaltspufferung, verwenden 63
JavaScript-Erweiterungen 77
Kontrollkästchen, Werte prüfen in 71
Layoutinformation, verwenden 63
Namespaces für 55
Query-String-Referenz 89
Query-Strings für 54
Seiten aus XSLT-Stylesheet generieren 44
Server-seitige Stylesheets 43, 51
Vergleich von Strings über reguläre Perl-Ausdrücke 71
Verwenden in Stylesheets in Website oder Programm 48
XSLT 1.0-Spezifikation 43
XSLT-CWP-Abfragen 44
XSLT-Site-Assistent, verwenden 46
xslt-template-files (Ordner) 45, 48, 63
Zusammenfassung der Schritte für Publishing 45
XSLT Custom Web Publishing 13
<?xslt-cwp-buffer buffer-content="true"?> (Verarbeitungsanweisung) 64
<?xslt-cwp-query?> (Verarbeitungsanweisung) 52, 56
XSLT-Site-Assistent
Beschreibung 46
Generierte Stylesheets, Beschreibung 48
Starten 47
Verwenden 47
Verwendung vorbereiten 46
xslt-template-files (Ordner) 45, 48, 63

Z

Zeit-Erweiterungsfunktionen, verwenden 72
Zeitformatstrings 74
Zugriffsprotokolldateien für Web-Server, Beschreibung 84
Zugriffsrechte 20
Zusammenfassung der Schritte für
XML-Datenzugriff 27
XSLT Publishing 45